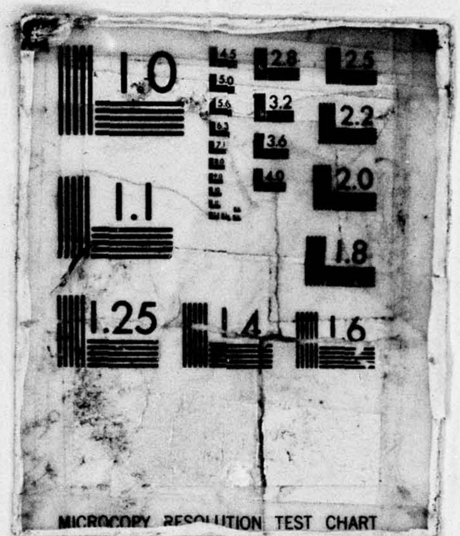
AD-A075 289 NUMERICAL ELECTROMAGNETIC CODE (NEC)-METHOD OF MOMENTS 1/6
A USER-ORIENTED CO.. (U) NAVAL OCEAN SYSTEMS CENTER SAN
DIEGO CA G J BURKE ET AL. 18 JUL 77 NOSC/TD-116-VOL-1
UNCLASSIFIED AFWL-TR-76-320-VOL-1 F/G 12/5 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART

NOSC

NOSC TD 116
Vol 1

NOSC TD 116
Vol 1

Technical Document 116
Volume 1

NUMERICAL ELECTROMAGNETICS CODE (NEC) - METHOD OF MOMENTS

A user-oriented computer code for analysis of the electromagnetic response of antennas and other metal structures

Part I: Program Description-Theory

Part II: Program Description-Code

(Vol 2 contains Part III: User's Guide)

GJ Burke and AJ Pogio
(Lawrence Livermore Laboratory)

January 1981

Prepared by
LAWRENCE LIVERMORE LABORATORY
(Report UCID 18834)

Prepared for
NAVAL ELECTRONIC SYSTEMS COMMAND
(NAVELEX 3041)

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

[redacted]

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

Work was performed by GJ Burke and AJ Poggio, of the Lawrence Livermore Laboratory, under MIPR-N0095376MP, as part of Program Element 62543N, Project XF43453401 (NOSC 8105-CM41) for Naval Electronic Systems Command, NAVELEX 3041. This document covers work up to September 1980.

Released by
MS Kvigne, Head
Communications Research and
Technology Division

Under authority of
HD Smith, Head
Communications Systems and
Technology Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Document 116 (TD 116) vol 1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NUMERICAL ELECTROMAGNETICS CODE (NEC) – METHOD OF MOMENTS; A user-oriented computer code for analysis of the electromagnetic response of antennas and other metal structures		5. TYPE OF REPORT & PERIOD COVERED Technical document Up to September 1980 6. PERFORMING ORG. REPORT NUMBER UCID 18834
7. AUTHOR(s) GJ Burke and AJ Poggio Lawrence Livermore Laboratory		8. CONTRACT OR GRANT NUMBER(s) MIPR-N0095376MP
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center (Code 8112) San Diego CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 62543N, Project XF43453401 (NOSC 8105-CM41)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command NAVELEX 3041 Washington DC 20360		12. REPORT DATE January 1981 – latest revision 13. NUMBER OF PAGES (as revised) Part I–81; Part II–433
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Volume 1 contains Part I: Program Description – Theory (revised 2 January 1980)

Part II: Program Description – Code (revised January 1981)

Volume 2 contains Part III: User's Guide (revised 2 January 1980)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Numerical Electromagnetics Code (NEC-2)
Numerical analysis
Antenna response
Electromagnetic radiation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Numerical Electromagnetics Code (NEC-2) is a computer code for analyzing the electromagnetic response of an arbitrary structure consisting of wires and surfaces in free space or over a ground plane. The analysis is accomplished by the numerical solution of integral equations for induced currents. The solution includes a Numerical Green's Function for partitioned-matrix solution and a treatment for lossy grounds that is accurate for antennas very close to the ground surface. The excitation may be an incident plane wave or a voltage source on a wire, while the output may include current and charge density, electric or magnetic field in the vicinity of the structure, and radiated fields. Other options compute the maximum coupling between antennas and facilitate

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

structure input. Hence the code may be used for antenna analysis, EMP, or scattering studies.

Part I of the document includes the equations on which the code is based and a discussion of the approximations and numerical methods used in the *numerical solution*. Some comparisons to demonstrate the range of accuracy of approximations are also included. Details of the coding and a User's Guide are provided as parts II and III, respectively.

Technical Document 116

NUMERICAL ELECTROMAGNETIC CODE (NEC) – METHOD OF MOMENTS

Part I: Program Description–Theory

GJ Burke and AJ Poggio

18 July 1977

Prepared for
NAVAL ELECTRONIC SYSTEMS COMMAND (ELEX 3041)

Completed

January 1981

INSTRUCTIONS FOR REVISION OF TD 116

1. Slip the first Volume 2 title page into the outside front pocket of the Volume 2 binder. Discard the old binder title page. (Note: The original issue of TD 116 included a set of white binders with pockets for tle pages. Subsequent copies issued may not have included the bir s.)

2. Insert the following pages at the beginning of Volume 2 in front of the title page for Part III:

<u>insert</u>	<u>page description</u>
(a) ____	Volume 2 title page (side 1) and Administrative Information page (side 2)
(b) ____	Report Documentation Page, DD Form 1473 (2 sides)
(c) ____	Volume 2 Revision Page (side 1; side 2 is blank)

3. Slip the first Volume 1 title page into the outside front pocket of the Volume 1 binder. (See note in step 1 above.)

4. Insert the following pages at the beginning of Volume 1 in front of the title page for Part I:

<u>insert</u>	<u>page description</u>
(a) ____	Volume 1 title page (side 1) and Administrative Information page (side 2)
(b) ____	Report Documentation Page, DD Form 1473 (2 sides)
(c) ____	Volume 1 Revision Page (side 1; side 2 is blank)

5. Insert all the remaining revision pages (listed below) in Volume 1 after page 81 of Part I.

<u>insert</u>	<u>page description</u>
(a) ____	Part II title page (side 1; side 2 is blank)
(b) ____	Pages i through vii
(c) ____	Pages 1 through 433

6. This revision replaces all pages of Part II. Remove and discard all <u>old</u> pages of the <u>old</u> Part II.

VOLUME 1 REVISION PAGE

This volume contains

Part I - Revised 2 January 1980

Part II - Revised January 1981

Part I: Program Description - Theory

(Revised 2 Jan 80)

Preface

The Numerical Electromagnetics Code (NEC) has been developed at the Lawrence Livermore Laboratory, Livermore, California, under the sponsorship of the Naval Ocean Systems Center and the Air Force Weapons Laboratory. It is an advanced version of the Antenna Modeling Program (AMP) developed in the early 1970's by MBAssociates for the Naval Research Laboratory, Naval Ship Engineering Center, U.S. Army ECOM/Communications Systems, U.S. Army Strategic Communications Command, and Rome Air Development Center under Office of Naval Research Contract N00014-71-C-0187. The present version of NEC is the result of efforts by G. J. Burke and A. J. Poggio of Lawrence Livermore Laboratory.

The documentation for NEC consists of three volumes:

Part I: NEC Program Description — Theory

Part II: NEC Program Description — Code

Part III: NEC User's Guide

The documentation has been prepared by using the AMP documents as foundations and by modifying those as needed. In some cases this led to minor changes in the original documents while in many cases major modifications were required.

Over the years many individuals have been contributors to AMP and NEC and are acknowledged here as follows:

R. W. Adams	R. J. Lytle
J. N. Brittingham	E. K. Miller
G. J. Burke	J. B. Morton
F. J. Deadrick	G. M. Pjerrou
K. K. Hazard	A. J. Poggio
D. L. Knepp	E. S. Selden
D. L. Lager	

The support for the development of NEC-2 at the Lawrence Livermore Laboratory has been provided by the Naval Ocean Systems Center under MIPR-N0095376MP. Cognizant individuals under whom this project was carried out include: J. Rockway and J. Logan. Previous development of NEC also included the support of the Air Force Weapons Laboratory (Project Order 76-090) and was monitored by J. Castillo and TSgt. H. Goodwin.

Work was performed under the auspices of the U.S. Department of Energy under contract No. W-7405-Eng-48. Reference to a company or product name

does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

Contents

List of Illustrations

Section I
Introduction

The Numerical Electromagnetics Code (NEC-2) is a user-oriented computer code for the analysis of the electromagnetic response of antennas and other metal structures. It is built around the numerical solution of integral equations for the currents induced on the structure by sources or incident fields. This approach avoids many of the simplifying assumptions required by other solution methods and provides a highly accurate and versatile tool for electromagnetic analysis.

The code combines an integral equation for smooth surfaces with one specialized to wires to provide for convenient and accurate modeling of a wide range of structures. A model may include nonradiating networks and transmission lines connecting parts of the structure, perfect or imperfect conductors, and lumped-element loading. A structure may also be modeled over a ground plane that may be either a perfect or imperfect conductor.

The excitation may be either voltage sources on the structure or an incident plane wave of linear or elliptic polarization. The output may include induced currents and charges, near electric or magnetic fields, and radiated fields. Hence, the program is suited to either antenna analysis or scattering and EMP studies. NEC and its predecessor AMP have been used successfully to model a wide range of antennas including complex environments such as ships. Results from modeling several antennas with NEC are shown in refs. 36, 37, and 38 with measured data for comparison.

The integral-equation approach is best suited to structures with dimensions up to several wavelengths. Although there is no theoretical size limit, the numerical solution requires a matrix equation of increasing order as the structure size is increased relative to wavelength. Hence, modeling very large structures may require more computer time and file storage than is practical on a particular machine. In such cases standard high-frequency approximations such as geometrical or physical optics, or geometric theory of diffraction may be more suitable than the integral equation approach used in NEC.

The code NEC-2 is the latest in a series of electromagnetics codes, each of which has built upon the previous one. The first in the series was the code BRACT which was developed at MBAssociates in San Ramon, California, under the funding of the Air Force Space and Missiles Systems Organization (refs. 1 and 2). BRACT was specialized to scattering by arbitrary thin-wire configurations.

The code AMP followed BRACT and was developed at MBAssociates with funding from the Naval Research Laboratory, Naval Ship Engineering Center, U.S. Army ECOM/Communications Systems, U.S. Army Strategic Communications Command, and Rome Air Development Center under Office of Naval Research Contract N00014-71-C-0187. AMP uses the same numerical solution method as BRACT with the addition of the capability of modeling a structure over a ground plane and an option to use file storage to greatly increase the maximum structure size that may be modeled. The program input and output were extensively revised for AMP so that the code could be used with a minimum of learning and computer programming experience. AMP includes extensive documentation to aid in understanding, using, and modifying the code (refs. 3, 4 and 5).

A modeling option specialized to surfaces was added to the wire modeling capabilities of AMP in the AMP2 code (ref. 6). A simplified approximation for large interaction distances was also included in AMP2 to reduce running time for large structures.

The code NEC-1 added to AMP2 a more accurate current expansion along wires and at multiple wire junctions, and an option in the wire modeling technique for greater accuracy on thick wires. A new model for a voltage source was added and several other modifications made for increased accuracy and efficiency.

NEC-2 retains all features of NEC-1 except for a restart option. Major additions in NEC-2 are the Numerical Green's Function for partitioned-matrix solution and a treatment for lossy grounds that is accurate for antennas very close to the ground surface. NEC-2 also includes an option to compute maximum coupling between antennas and new options for structure input.

Part I of this document describes the equations and numerical methods used in NEC. Part III: NEC User's Guide (ref. 7) contains instructions for using the code, including preparation of input and interpretation of output. Part II: NEC Program Description — Code (ref. 8) describes the coding in detail. The user encountering the code for the first time should begin with the User's Guide and try modeling some simple antennas. Part II will be of interest mainly to someone attempting to modify the code. Reading part I will be useful to the new user of NEC-2, however, since an understanding of the theory and solution method will assist in the proper application of the code.

Section II
The Integral Equations For Free Space

The NEC program uses both an electric-field integral equation (EFIE) and a magnetic-field integral equation (MFIE) to model the electromagnetic response of general structures. Each equation has advantages for particular structure types. The EFIE is well suited for thin-wire structures of small or vanishing conductor volume while the MFIE, which fails for the thin-wire case, is more attractive for voluminous structures, especially those having large smooth surfaces. The EFIE can also be used to model surfaces and is preferred for thin structures where there is little separation between a front and back surface. Although the EFIE is specialized to thin wires in this program, it has been used to represent surfaces by wire grids with reasonable success for far-field quantities but with variable accuracy for surface fields. For a structure containing both wires and surfaces the EFIE and MFIE are coupled. This combination of the EFIE and MFIE was proposed and used by Albertsen, Hansen, and Jensen at the Technical University of Denmark (ref. 9) although the details of their numerical solution differ from those in NEC. A rigorous derivation of the EFIE and MFIE used in NEC is given by Poggio and Miller (ref. 10). The equations and their derivation are outlined in the following sections.

1. THE ELECTRIC FIELD INTEGRAL EQUATION (EFIE)

The form of the EFIE used in NEC follows from an integral representation for the electric field of a volume current distribution $\vec{J}$,

$$\vec{E}(\vec{r}) = \frac{-j\eta}{4\pi k} \int_V \vec{J}(\vec{r}\,') \cdot \overline{\overline{G}}(\vec{r}, \vec{r}\,')\, dV', \tag{1}$$

where

$$\overline{\overline{G}}(\vec{r}, \vec{r}\,') = (k^2\overline{\overline{I}} + \nabla\nabla)g(\vec{r}, \vec{r}\,'),$$

$$g(\vec{r}, \vec{r}\,') = \exp(-jk|\vec{r} - \vec{r}\,'|)/|\vec{r} - \vec{r}\,'|,$$

$$k = \omega\sqrt{\mu_o\varepsilon_o}\ ,$$

$$\eta = \sqrt{\mu_o/\varepsilon_o}$$

and the time convention is exp(jωt). $\overline{\overline{I}}$ is the identity dyad $(\hat{x}\hat{x} + \hat{y}\hat{y} + \hat{z}\hat{z})$. When the current distribution is limited to the surface of a perfectly conducting body, equation (1) becomes

$$\vec{E}(\vec{r}) = \frac{-j\eta}{4\pi k} \int_S \vec{J}_s(\vec{r}\,') \cdot \overline{\overline{G}}(\vec{r}, \vec{r}\,')\, dA', \tag{2}$$

with $\vec{J}_s$ the surface current density. The observation point $\vec{r}$ is restricted to be off the surface S so that $\vec{r} \neq \vec{r}\,'$.

If $\vec{r}$ approaches S as a limit, equation (2) becomes

$$\vec{E}(\vec{r}) = \frac{-j\eta}{4\pi k} ⨍_S \vec{J}_s(\vec{r}\,') \cdot \overline{\overline{G}}(\vec{r}, \vec{r}\,')\, dA', \tag{3}$$

where the principal value integral, $⨍$, is indicated since $g(\vec{r}, \vec{r}\,')$ is now unbounded.

An integral equation for the current induced on S by an incident field $\vec{E}^I$ can be obtained from equation (3) and the boundary condition for $\vec{r} \in S$,

$$\hat{n}(\vec{r}) \times \left[\vec{E}^s(\vec{r}) + \vec{E}^I(\vec{r})\right] = 0, \tag{4}$$

where $\hat{n}(\vec{r})$ is the unit normal vector of the surface at $\vec{r}$ and $\vec{E}^s$ is the field due to the induced current $\vec{J}_s$. Substituting equation (3) for $\vec{E}^s$ yields the integral equation,

$$-\hat{n}(\vec{r}) \times \vec{E}^I(\vec{r}) = \frac{-j\eta}{4\pi k}\, \hat{n}(\vec{r}) \times$$

$$⨍_S \vec{J}_s(\vec{r}\,') \cdot (k^2\overline{\overline{I}} + \nabla\nabla) g(\vec{r}, \vec{r}\,')\, dA'. \tag{5}$$

The vector integral in equation (5) can be reduced to a scalar integral equation when the conducting surface S is that of a cylindrical thin wire, thereby making the solution much easier. The assumptions applied for a thin wire, known as the thin-wire approximation, are as follows:

a. Transverse currents can be neglected relative to axial currents on the wire.

b. The circumferential variation in the axial current can be neglected.

c. The current can be represented by a filament on the wire axis.

d. The boundary condition on the electric field need be enforced in the axial direction only.

These widely used approximations are valid as long as the wire radius is much less than the wavelength and much less than the wire length. An alternate kernel for the EFIE, based on an extended thin-wire approximation in which condition c is relaxed, is also included in NEC for wires having too large a radius for the thin-wire approximation (ref. 11).

From assumptions a, b and c, the surface current $\vec{J}_s(\vec{r})$ on a wire of radius a can be replaced by a filamentary current I where

$$I(s)\hat{s} = 2\pi a \vec{J}_s(\vec{r}),$$

s = distance parameter along the wire axis at $\vec{r}$, and

$\hat{s}$ = unit vector tangent to the wire axis at $\vec{r}$.

Equation (5) then becomes

$$-\hat{n}(\vec{r}) \times \vec{E}^I(\vec{r}) = \frac{-j\eta}{4\pi k}\,\hat{n}(\vec{r}) \times \int_L I(s')\left(k^2\hat{s}' - \nabla\frac{\partial}{\partial s'}\right) g(\vec{r}, \vec{r}')\, ds', \tag{6}$$

where the integration is over the length of the wire. Enforcing the boundary condition in the axial direction reduces Eq. (6) to the scalar equation,

$$-\hat{s} \cdot \vec{E}^I(\vec{r}) = \frac{-j\eta}{4\pi k}\int_L I(s')\left(k^2\hat{s}\cdot\hat{s}' - \frac{\partial^2}{\partial s\,\partial s'}\right) g(\vec{r}, \vec{r}')\, ds'. \tag{7}$$

Since $\vec{r}'$ is now the point at s' on the wire axis while $\vec{r}$ is a point at s on the wire surface $|\vec{r} - \vec{r}'| \geq a$ and the integrand is bounded.

2. THE MAGNETIC FIELD INTEGRAL EQUATION (MFIE)

The MFIE is derived from the integral representation for the magnetic field of a surface current distribution $\vec{J}_s$,

$$\vec{H}^s(\vec{r}) = \frac{1}{4\pi} \int_S \vec{J}_s(\vec{r}\,') \times \nabla' g(\vec{r}, \vec{r}\,')\, dA', \tag{8}$$

where the differentiation is with respect to the integration variable $\vec{r}\,'$. If the current $\vec{J}_s$ is induced by an external incident field $\vec{H}^I$, then the total magnetic field inside the perfectly conducting surface must be zero. Hence, for $\vec{r}$ just inside the surface S,

$$\vec{H}^I(\vec{r}) + \vec{H}^s(\vec{r}) = 0, \tag{9}$$

where $\vec{H}^I$ is the incident field with the structure removed, and $\vec{H}^s$ is the scattered field given by equation (8). The integral equation for $\vec{J}_s$ may be obtained by letting $\vec{r}$ approach the surface point $\vec{r}_o$ from inside the surface along the normal $\hat{n}(\vec{r}_o)$. The surface component of equation (9) with equation (8) substituted for $\vec{H}^s$ is then

$$-\hat{n}(\vec{r}_o) \times \vec{H}^I(\vec{r}_o) = \hat{n}(\vec{r}_o) \times \frac{1}{4\pi} \lim_{\vec{r} \to \vec{r}_o} \int_S \vec{J}_s(\vec{r}\,') \times \nabla' g(\vec{r}, \vec{r}\,')\, dA',$$

where $\hat{n}(\vec{r}_o)$ is the outward directed normal vector at $\vec{r}_o$. The limit can be evaluated by using a result of potential theory (ref. 12) to yield the integral equation

$$-\hat{n}(\vec{r}_o) \times \vec{H}^I(\vec{r}_o) = -\frac{1}{2} \vec{J}_s(\vec{r}_o) + \frac{1}{4\pi} \int_S \hat{n}(\vec{r}_o) \times \left[\vec{J}_s(\vec{r}\,') \times \nabla' g(\vec{r}_o, \vec{r}\,')\right] dA'. \tag{10}$$

For solution in NEC, this vector integral equation is resolved into two scalar equations along the orthogonal surface vectors $\hat{t}_1$ and $\hat{t}_2$ where

$$\hat{t}_1(\vec{r}_o) \times \hat{t}_2(\vec{r}_o) = \hat{n}(\vec{r}_o).$$

By using the identity $\vec{u} \cdot (\vec{v} \times \vec{w}) = (\vec{u} \times \vec{v}) \cdot \vec{w}$ and noting that $\hat{t}_1 \times \hat{n} = -\hat{t}_2$ and $\hat{t}_2 \times \hat{n} = \hat{t}_1$, the scalar equations can be written,

$$\hat{t}_2(\vec{r}_o) \cdot \vec{H}^I(\vec{r}_o) = -\frac{1}{2}\,\hat{t}_1(\vec{r}_o) \cdot \vec{J}_s(\vec{r}_o) - \frac{1}{4\pi} \int_S \hat{t}_2(\vec{r}_o) \cdot \left[\vec{J}_s(\vec{r}\,') \times \nabla' g(\vec{r}_o, \vec{r}\,')\right] dA'; \tag{11}$$

$$-\hat{t}_1(\vec{r}_o) \cdot \vec{H}^I(\vec{r}_o) = -\frac{1}{2}\,\hat{t}_2(\vec{r}_o) \cdot \vec{J}_s(\vec{r}_o) + \frac{1}{4\pi} \int_S \hat{t}_1(\vec{r}_o) \cdot \left[\vec{J}_s(\vec{r}\,') \times \nabla' g(\vec{r}_o, \vec{r}\,')\right] dA'. \tag{12}$$

These two components suffice since there is no normal component of equation (10).

3. THE EFIE-MFIE HYBRID EQUATION

Program NEC uses the EFIE for thin wires and the MFIE for surfaces. For a structure consisting of both wires and surfaces, $\vec{r}$ in equation (7) is restricted to the wires, with the integral for $E^s(\vec{r})$, extending over the complete structure. The thin-wire form of the integral in equation (7) is used over wires while the more general form of equation (5) must be used on surfaces. Likewise, $\vec{r}_o$ is restricted to surfaces in equations (11) and (12), with the integrals for $H^s(\vec{r})$ extending over the complete structure. On wires the integral is simplified by the thin-wire approximation. The resulting coupled integral equations are, for $\vec{r}$ on wire surfaces,

$$-\hat{s} \cdot \vec{E}^I(\vec{r}) = \frac{-j\eta}{4\pi k} \int_L I(s')\left(k^2\hat{s} \cdot \hat{s}' - \frac{\partial^2}{\partial s \partial s'}\right) g(\vec{r},\vec{r}\,')\, ds' - \frac{j\eta}{4\pi k} \int_{S_1} \vec{J}_s(\vec{r}) \cdot \left[k^2\hat{s} - \nabla' \frac{\partial}{\partial s}\right] g(\vec{r},\vec{r}\,')\, dA', \tag{13}$$

and for $\vec{r}$ on surfaces excluding wires

$$\hat{t}_2(\vec{r}) \cdot \vec{H}^I(\vec{r}) = -\frac{1}{4\pi}\hat{t}_2(\vec{r}) \cdot \int_L I(s')\left(\hat{s}' \times \nabla' g(\vec{r},\vec{r}')\right) ds'$$

$$-\frac{1}{2}\hat{t}_1(\vec{r}) \cdot \vec{J}_s(\vec{r}) -$$

$$\frac{1}{4\pi}\oint_{S_1} \hat{t}_2(\vec{r}) \cdot \left[\vec{J}_s(\vec{r}') \times \nabla' g(\vec{r},\vec{r}')\right] dA', \quad (14)$$

and

$$-\hat{t}_1(\vec{r}) \cdot \vec{H}^I(\vec{r}) = \frac{1}{4\pi}\hat{t}_1(\vec{r}) \cdot \int_L I(s')\left(s' \times \nabla' g(\vec{r},\vec{r}')\right) ds'$$

$$-\frac{1}{2}\hat{t}_2(\vec{r}) \cdot \vec{J}_s(\vec{r}) +$$

$$\frac{1}{4\pi}\oint_{S_1} \hat{t}_1(\vec{r}) \cdot \left[\vec{J}_s(\vec{r}') \times \nabla' g(\vec{r},\vec{r}')\right] dA'. \quad (15)$$

The symbol $\int_L$ represents integration over wires while $\int_{S_1}$ represents integration over surfaces excluding wires. The numerical method used to solve equations (13), (14) and (15) is described in section III.

Section III
Numerical Solution

The integral equations (13), (14), and (15) are solved numerically in NEC by a form of the method of moments. An excellent general introduction to the method of moments can be found in R. F. Harrington's book, Field Computation by Moment Methods (ref. 13). A brief outline of the method follows.

The method of moments applies to a general linear-operator equation,

$$Lf = e, \tag{16}$$

where f is an unknown response, e is a known excitation, and L is a linear operator (an integral operator in the present case). The unknown function f may be expanded in a sum of basis functions, f_j, as

$$f = \sum_{j=1}^{N} \alpha_j f_j \; . \tag{17}$$

A set of equations for the coefficients α_j are then obtained by taking the inner product of equation (16) with a set of weighting functions $\{w_i\}$,

$$\langle w_i, Lf \rangle = \langle w_i, e \rangle \qquad i = 1, \ldots N \; . \tag{18}$$

Due to the linearity of L equation (17) substituted for f yields,

$$\sum_{j=1}^{N} \alpha_j \langle w_i, Lf_j \rangle = \langle w_i, e \rangle \; , \qquad i = 1, \ldots N \; .$$

This equation can be written in matrix notation as

$$[G]\,[A] = [E] \; , \tag{19}$$

where

$$G_{ij} = \langle w_i, Lf_j \rangle \; ,$$

$$A_j = \alpha_j \; ,$$

$$E_i = \langle w_i, e \rangle \; .$$

The solution is then

$$[A] = [G]^{-1} [E] .$$

For the solution of equations (13), (14), and (15), the inner product is defined as

$$<f, g> = \int_S f(\vec{r})g(\vec{r})dA ,$$

where the integration is over the structure surface. Various choices are possible for the weighting functions $\{w_i\}$ and basis functions $\{f_j\}$. When $w_i = f_i$, the procedure is known as Galerkin's method. In NEC the basis and weight functions are different, w_i being chosen as a set of delta functions

$$w_i(\vec{r}) = \delta(\vec{r} - \vec{r}_i) ,$$

with $\{\vec{r}_i\}$ a set of points on the conducting surface. The result is a point sampling of the integral equations known as the collocation method of solution. Wires are divided into short straight segments with a sample point at the center of each segment while surfaces are approximated by a set of flat patches or facets with a sample point at the center of each patch.

The choice of basis functions is very important for an efficient and accurate solution. In NEC the support of f_i is restricted to a localized subsection of the surface near $\vec{r}_i$. This choice simplifies the evaluation of the inner-product integral and ensures that the matrix G will be well conditioned. For finite N, the sum of f_j cannot exactly equal a general current distribution so the functions f_i should be chosen as close as possible to the actual current distribution. Because of the nature of the integral-equation kernels, the choice of basis function is much more critical on wires than on surfaces. The functions used in NEC are explained in the following sections.

1. CURRENT EXPANSION ON WIRES

Wires in NEC are modeled by short straight segments with the current on each segment represented by three terms — a constant, a sine, and a cosine. This expansion was first used by Yeh and Mei (ref. 14) and has been shown to provide rapid solution convergence (ref. 15 and 16). It has the added advantage that the fields of the sinusoidal currents are easily evaluated in closed form. The amplitudes of the constant, sine, and cosine terms are related such

that their sum satisfies physical conditions on the local behavior of current and charge at the segment ends. This differs from AMP where the current was extrapolated to the centers of the adjacent segments, resulting in discontinuities in current and charge at the segment ends. Matching at the segment ends improves the solution accuracy, especially at the multiple-wire junctions of unequal length segments where AMP extrapolated to an average length segment, often with inaccurate results.

The total current on segment number j in NEC has the form

$$I_j(s) = A_j + B_j \sin k(s-s_j) + C_j \cos k(s-s_j) \,, \tag{20}$$

$$|s-s_j| < \Delta_j/2 \,,$$

where s_j is the value of s at the center of segment j and Δ_j is the length of segment j. Of the three unknown constants A_j, B_j, and C_j, two are eliminated by local conditions on the current leaving one constant, related to the current amplitude, to be determined by the matrix equation. The local conditions are applied to the current and to the linear charge density, q, which is related to the current by the equation of continuity

$$\frac{\partial I}{\partial s} = - j\omega q \,. \tag{21}$$

At a junction of two segments with uniform radius, the obvious conditions are that the current and charge are continuous at the junction. At a junction of two or more segments with unequal radii, the continuity of current is generalized to Kirchoff's current law that the sum of currents into the junction is zero. The total charge in the vicinity of the junction is assumed to distribute itself on individual wires according to the wire radii, neglecting local coupling effects. T. T. Wu and R. W. P. King (ref. 17) have derived a condition that the linear charge density on a wire at a junction, and hence $\partial I/\partial s$, is determined by

$$\left.\frac{\partial I(s)}{\partial s}\right|_{s \text{ at junction}} = \frac{Q}{\ell n\left(\frac{2}{ka}\right) - \gamma} \,, \tag{22}$$

where a = wire radius,

$$k = 2\pi/\lambda \,,$$

$\gamma = 0.5772$ (Euler's constant).

Q is related to the total charge in the vicinity of the junction and is constant for all wires at the junction.

At a free wire end, the current may be assumed to go to zero. On a wire of finite radius, however, the current can flow onto the end cap and hence be nonzero at the wire end. In one study of this effect, a condition relating the current at the wire end to the current derivative was derived (ref. 18). For a wire of radius a, this condition is

$$I(s)\bigg|_{s \text{ at end}} = \frac{-(\hat{s}\cdot\hat{n}_c)}{k}\,\frac{J_1(ka)}{J_o(ka)}\,\frac{\partial I(s)}{\partial s}\bigg|_{s \text{ at end}},$$

where J_o and J_1 are Bessel functions of order 0 and 1. The unit vector $\hat{n}_c$ is normal to the end cap. Hence, $\hat{s} \cdot \hat{n}_c$ is +1 if the reference direction, $\hat{s}$, is toward the end, and -1 if $\hat{s}$ is away from the end.

Thus, for each segment two equations are obtained from the two ends:

$$I_j(s_j \pm \Delta_j/2) = \frac{\pm 1}{k}\,\frac{J_1(ka_j)}{J_o(ka_j)}\,\frac{\partial I_j(s)}{\partial s}\bigg|_{s = s_j \pm \Delta_j/2} \tag{23}$$

at free ends, and

$$\frac{\partial I_j(s)}{\partial s}\bigg|_{s = s_j \pm \Delta_j/2} = \frac{Q_j^{\pm}}{\ln\left(\frac{2}{ka_j}\right) - \gamma} \tag{24}$$

at junctions. Two additional unknowns Q_j^- and Q_j^+ are associated with the junctions but can be eliminated by Kirchoff's current equation at each junction. The boundary-condition equations provide the additional equation-per-segment to completely determine the current function of equation (20) for every segment.

To apply these conditions, the current is expanded in a sum of basis functions chosen so that they satisfy the local conditions on current and charge in any linear combination. A typical set of basis functions and their sum on a four segment wire are shown in figure 1. For a general segment i in figure 2, the ith basis function has a peak on segment i and extends onto

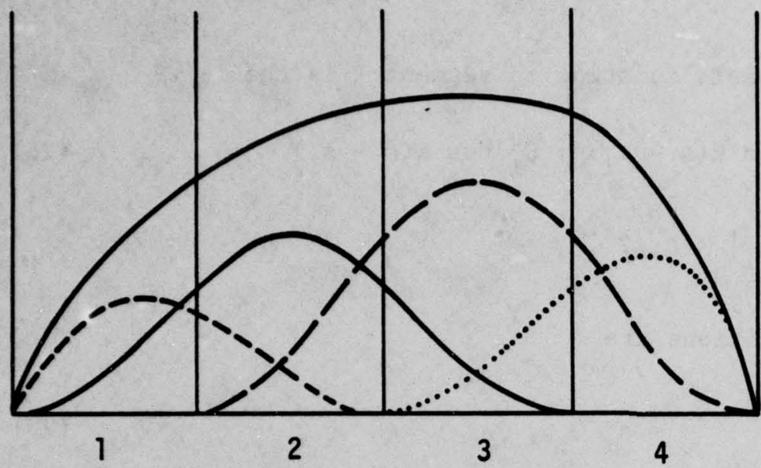


Figure 1. Current Basis Functions and Sum on a Four Segment Wire.

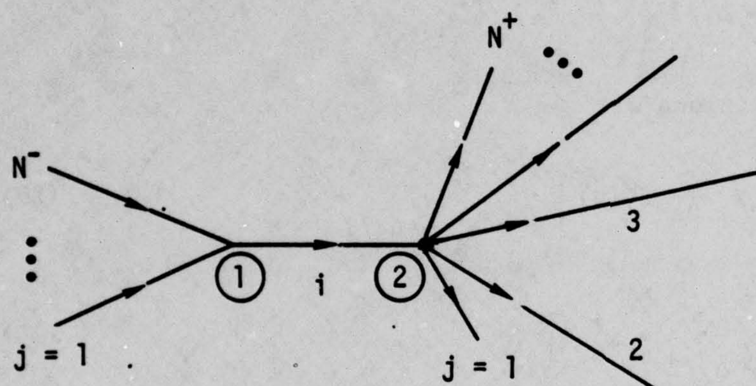


Figure 2. Segments Covered by the i^{th} Basis Function.

every segment connected to i, going to zero with zero derivative at the outer ends of the connected segments.

The general definition of the i^{th} basis function is given below. For the junction and end conditions described above, the following definitions apply for the factors in the segment end conditions:

$$a_i^- = a_i^+ = \left[\ell n \left(\frac{2}{ka_i}\right) - \gamma\right]^{-1} , \tag{25}$$

and

$$X_i = J_1(ka_i)/J_o(ka_i) .$$

The condition of zero current at a free end may be obtained by setting X_i to zero.

The portion of the i^{th} basis function on segment i is then

$$f_i^o(s) = A_i^o + B_i^o \sin k(s - s_i) + C_i^o \cos k(s - s_i) \tag{26}$$

$$|s - s_i| < \Delta_i/2 \; .$$

If $N^- \neq 0$ and $N^+ \neq 0$, end conditions are

$$\frac{\partial}{\partial s} f_i^o(s) \Big|_{s = s_i - \Delta_i/2} = a_i^- Q_i^- \; , \tag{27}$$

$$\frac{\partial}{\partial s} f_i^o(s) \Big|_{s = s_i + \Delta_i/2} = a_i^+ Q_i^+ \; . \tag{28}$$

If $N^- = 0$ and $N^+ \neq 0$, end conditions are

$$f_i^o(s_i - \Delta_i/2) = \frac{1}{k} X_i \frac{\partial}{\partial s} f_i^o(s) \Big|_{s = s_i - \Delta_i/2} \tag{29}$$

$$\frac{\partial}{\partial s} f_i^o(s) \Big|_{s = s_i + \Delta_i/2} = a_i^+ Q_i^+ \; . \tag{30}$$

If $N^- \neq 0$ and $N^+ = 0$, end conditions are

$$\frac{\partial}{\partial s} f_i^o(s) \Big|_{s = s_i - \Delta_i/2} = a_i^- Q_i^- \; , \tag{31}$$

$$f_i^o(s_i + \Delta_i/2) = \frac{-1}{k} X_i \frac{\partial}{\partial s} f_i^o(s) \Big|_{s = s_i + \Delta_i/2} \; . \tag{32}$$

Over segments connected to end 1 of segment i, the i^{th} basis function is

$$f_j^-(s) = A_j^- + B_j^- \sin k(s - s_j) + C_j^- \cos k(s - s_j) \tag{33}$$

$$|s - s_j| < \Delta_j/2 \quad j = 1, \ldots, N^- \; .$$

End conditions are

$$f_j^-(s_j - \Delta_j/2) = 0 \ , \tag{34}$$

$$\frac{\partial}{\partial s} f_j^-(s) \Big|_{s = s_j - \Delta_j/2} = 0 \ , \tag{35}$$

$$\frac{\partial}{\partial s} f_j^-(s) \Big|_{s = s_j + \Delta_j/2} = a_j^+ Q_i^- \ . \tag{36}$$

Over segments connected to end 2 of segment i, the i^{th} basis function is

$$f_j^+(s) = A_j^+ + B_j^+ \sin k(s - s_j) + C_j^+ \cos k(s - s_j) \tag{37}$$

$$|s - s_j| < \Delta_j/2 \quad j = 1, \ldots, N^+ \ .$$

End conditions are

$$\frac{\partial}{\partial s} f_j^+(s) \Big|_{s = s_j - \Delta_j/2} = a_j^- Q_i^+ \ , \tag{38}$$

$$f_j^+(s_j + \Delta_j/2) = 0 \ , \tag{39}$$

$$\frac{\partial}{\partial s} f_j^+(s) \Big|_{s = s_j + \Delta_j/2} = 0 \ . \tag{40}$$

Equations (26), (33), and (37), defining the complete basis function, involve $3(N^- + N^+ + 1)$ unknown constants. Of these, $3(N^- + N^+) + 2$ unknowns are eliminated by the end conditions in terms of Q_i^- and Q_i^+ which can then be determined from the two Kirchoff's current equations:

$$\sum_{j=1}^{N^-} f_j^-(s_j + \Delta_j/2) = f_i^o(s_i - \Delta_i/2) \ , \text{ and} \tag{41}$$

$$\sum_{j=1}^{N^+} f_j^+(s_j - \Delta_j/2) = f_i^o(s_i + \Delta_i/2) \ . \tag{42}$$

The complete basis function is then defined in terms of one unknown constant. In this case A_i^o was set to -1 since the function amplitude is arbitrary, being determined by the boundary condition equations. The final result is given below:

$$A_j^- = \frac{a_j^+ Q_i^-}{\sin k \Delta_j} , \tag{43}$$

$$B_j^- = \frac{a_j^+ Q_i^-}{2 \cos k \Delta_j/2} , \tag{44}$$

$$C_j^- = \frac{- a_j^+ Q_i^-}{2 \sin k \Delta_j/2} , \tag{45}$$

$$A_j^+ = \frac{- a_j^- Q_i^+}{\sin k \Delta_j} , \tag{46}$$

$$B_j^+ = \frac{a_j^- Q_i^+}{2 \cos k \Delta_j/2} , \tag{47}$$

$$C_j^+ = \frac{a_j^- Q_i^+}{2 \sin k \Delta_j/2} . \tag{48}$$

For $N^- \neq 0$ and $N^+ \neq 0$,

$$A_i^o = - 1 , \tag{49}$$

$$B_i^o = \left(a_i^- Q_i^- + a_i^+ Q_i^+ \right) \frac{\sin k \Delta_i/2}{\sin k \Delta_i} , \tag{50}$$

$$C_i^o = \left(a_i^- Q_i^- - a_i^+ Q_i^+ \right) \frac{\cos k \Delta_i/2}{\sin k \Delta_i} , \tag{51}$$

$$Q_i^- = \frac{a_i^+(1 - \cos k\,\Delta_i) - P_i^+ \sin k\,\Delta_i}{\left(P_i^- P_i^+ + a_i^- a_i^+\right) \sin k\,\Delta_i + \left(P_i^- a_i^+ - P_i^+ a_i^-\right) \cos k\,\Delta_i} , \tag{52}$$

$$Q_i^+ = \frac{a_i^-(\cos k\,\Delta_i - 1) - P_i^- \sin k\,\Delta_i}{\left(P_i^- P_i^+ + a_i^- a_i^+\right) \sin k\,\Delta_i + \left(P_i^- a_i^+ - P_i^+ a_i^-\right) \cos k\,\Delta_i} . \tag{53}$$

For $N^- = 0$ and $N^+ \neq 0$,

$$A_i^o = -1 , \tag{54}$$

$$B_i^o = \frac{\sin k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} + a_i^+ Q_i^+ \frac{\cos k\,\Delta_i/2 - X_i \sin k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} , \tag{55}$$

$$C_i^o = \frac{\cos k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} + a_i^+ Q_i^+ \frac{\sin k\,\Delta_i/2 - X_i \cos k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} , \tag{56}$$

$$Q_i^+ = \frac{\cos k\,\Delta_i - 1 - X_i \sin k\,\Delta_i}{\left(a_i^+ + X_i P_i^+\right) \sin k\,\Delta_i + \left(a_i^+ X_i - P_i^+\right) \cos k\,\Delta_i} \tag{57}$$

For $N^- \neq 0$ and $N^+ = 0$,

$$A_i^o = -1 , \tag{58}$$

$$B_i^o = \frac{-\sin k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} + a_i^- Q_i^- \frac{\cos k\,\Delta_i/2 - X_i \sin k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} , \tag{59}$$

$$C_i^o = \frac{\cos k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} - a_i^- Q_i^- \frac{\sin k\,\Delta_i/2 + X_i \cos k\,\Delta_i/2}{\cos k\,\Delta_i - X_i \sin k\,\Delta_i} , \tag{60}$$

$$Q_i^- = \frac{1 - \cos k\,\Delta_i + X_i \sin k\,\Delta_i}{\left(a_i^- - X_i P_i^-\right) \sin k\,\Delta_i + \left(P_i^- + X_i a_i^-\right) \cos k\,\Delta_i} . \tag{61}$$

For all cases,

$$P_i^- = \sum_{j=1}^{N^-} \left(\frac{1 - \cos k\, \Delta_j}{\sin k\, \Delta_j} \right) a_j^+ , \tag{62}$$

$$P_i^+ = \sum_{j=1}^{N^+} \left(\frac{\cos k\, \Delta_j - 1}{\sin k\, \Delta_j} \right) a_j^- , \tag{63}$$

where the sum for P_i^- is over segments connected to end 1 of segment i, and the sum for P_i^+ is over segments connected to end 2. If $N^- = N^+ = 0$, the complete basis function is

$$f_i^o = \frac{\cos k(s - s_i)}{\cos k\, \Delta_i/2 - X_i \sin \Delta_i/2} - 1 . \tag{64}$$

When a segment end is connected to a ground plane or to a surface modeled with the MFIE, the end condition on both the total current and the last basis function is

$$\frac{\partial}{\partial s} I_j(s) \Big|_{s = s_j \pm \Delta_j/2} = 0 ,$$

replacing the zero current condition at a free end. This condition does not require a separate treatment, however, but is obtained by computing the last basis function as if the last segment is connected to its image segment on the other side of the surface.

It should be noted that in AMP, the basis function f_i has unit value at the center of segment i and zero value at the centers of connected segments although it does extend onto the connected segments. As a result, the amplitude of f_i is the total current at the center of segment i. This is not true in NEC so the current at the center of segment i must be computed by summing the contributions of all basis functions extending onto segment i.

2. CURRENT EXPANSION ON SURFACES

Surfaces on which the MFIE is used are modeled by small flat patches. The surface current on each patch is expanded in a set of pulse functions except in the region of wire connection, as will be described later. The pulse function expansion for N_p patches is

$$\vec{J}_s(\vec{r}) = \sum_{j=1}^{N_p} (J_{1j}\ \hat{t}_{1j} + J_{2j}\ \hat{t}_{2j})\ V_j(\vec{r})\ , \tag{65}$$

where

$$\hat{t}_{1j} = \hat{t}_1(\vec{r}_j)\ ,$$

$$\hat{t}_{2j} = \hat{t}_2(\vec{r}_j)\ ,$$

$\vec{r}_j$ = position of the center of patch number j ,

$V_j(\vec{r})$ = 1 for $\vec{r}$ on patch j and 0 otherwise.

The constants J_{1j} and J_{2j}, representing average surface-current density over the patch, are determined by the solution of the linear system of equations derived from the integral equations. The integrals for fields, due to the pulse basis functions, are evaluated numerically in a single step so that for integration, the pulses could be reduced to delta functions at the patch centers. That this simple approximation of the current yields good accuracy is one of the advantages of the MFIE for surfaces.

A more realistic representation of the surface current is needed, however, in the region where a wire connects to the surface. The treatment used in NEC, affecting the four coplanar patches about the connection point, is quite similar to that used by Albertsen et al. (ref. 9). In the region of the wire connection, the surface current contains a singular component due to the current flowing from the wire onto the surface. The total surface current should satisfy the condition,

$$\nabla_s \cdot \vec{J}_s(x,y) = J_o(x,y) + I_o\delta(x,y)\ ,$$

where the local coordinates x and y are defined in figure 3, ∇_s denotes surface divergence, $J_o(x,y)$ is a continuous function in the region ABCD, and I_o is the current at the base of the wire flowing onto the surface. One expansion which meets this requirement is

$$\vec{J}_s(x,y) = I_o\vec{f}(x,y) + \sum_{j=1}^{4} g_j(x,y)(\vec{J}_j - I_o\ \vec{f}_j)\ , \tag{66}$$

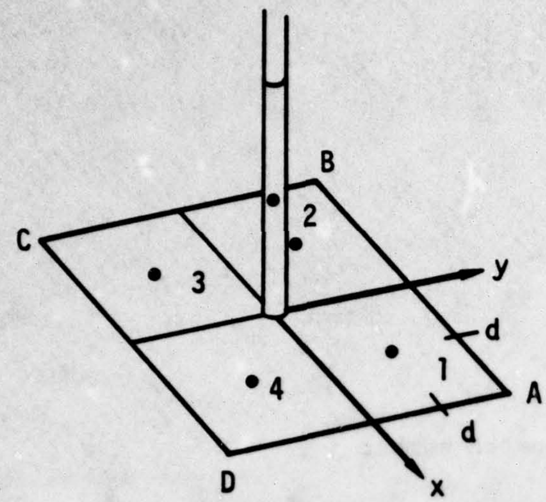


Figure 3. Detail of the Connection of a Wire to a Surface.

where

$$\vec{f}(x,y) = \frac{x\hat{x} + y\hat{y}}{2\pi(x^2 + y^2)} ,$$

$$\vec{J}_j = \vec{J}_s(x_j, y_j) ,$$

$$\vec{f}_j = \vec{f}(x_j, y_j) \text{ , and}$$

$(x_j, y_j) = (x,y)$ at the center of patch j. The interpolation functions $g_j(x,y)$ are chosen such that: $g_j(x,y)$ is differentiable on ABCD; $g_j(x_i,y_i) = \delta_{ij}$; and $\sum_{j=1}^{4} g_j(x,y) = 1$. The specific functions used in NEC are as follows:

$$g_1(x,y) = \frac{1}{4d^2}(d+x)(d+y) \qquad g_2(x,y) = \frac{1}{4d^2}(d-x)(d+y)$$

$$g_3(x,y) = \frac{1}{4d^2}(d-x)(d-y) \qquad g_4(x,y) = \frac{1}{4d^2}(d+x)(d-y)$$

Equation (66) is used when computing the electric field at the center of the connected wire segment due to the surface current on the four surrounding patches. In computing the field on any other segments or on any patches, the pulse-function form is used for all patches including those at the connection point. This saves integration time and is sufficiently accurate for the greater source to observation-point separations involved.

3. EVALUATION OF THE FIELDS

The current on each wire segment has the form

$$I_i(s) = A_i + B_i \sin k(s - s_i) + C_i \cos k(s - s_i) \qquad |s - s_i| < \Delta_i/2 , \tag{67}$$

where $k = \omega\sqrt{\mu_o \varepsilon_o}$, and Δ_i is the segment length. The solution requires the evaluation of the electric field at each segment due to this current. Three

approximations of the integral equation kernel are used: a thin-wire form for most cases, an extended thin-wire form for thick wires, and a current element approximation for large interaction distances. In each case the evaluation of the field is greatly simplified by the use of formulas for the fields of the constant and sinusoidal current components.

The accuracy of the thin-wire approximation for a wire of radius a and length Δ depends on ka and Δ/a. Studies have shown that the thin-wire approximation leads to errors of less than 1% for Δ/a greater than 8 (ref. 11). Furthermore, in the numerical solution of the EFIE, the wire is divided into segments less than about 0.1λ in length to obtain an adequate representation of current distribution thus restricting ka to less than about 0.08. The extended thin-wire approximation is applicable to shorter and thicker segments, resulting in errors less than 1% for Δ/a greater than 2.

For the thin-wire kernel, the source current is approximated by a filament on the segment axis while the observation point is on the surface of the observation segment. The fields are evaluated with the source segment on the axis of a local cylindrical-coordinate system as illustrated in figure 4.

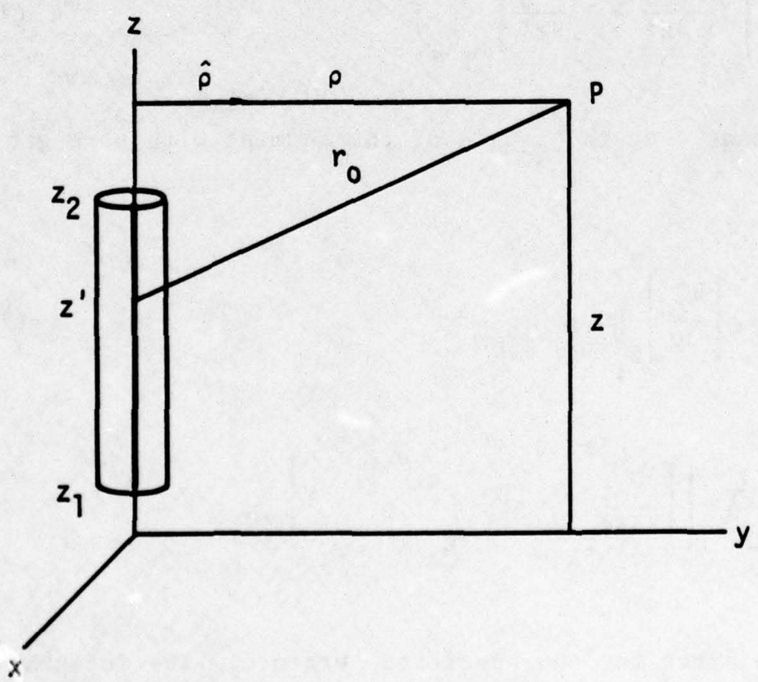


Figure 4. Current-Filament Geometry for the Thin-Wire Kernel.

Then with

$$G_o = \exp(-jkr_o)/r_o \ , \tag{68}$$

$$r_o = \left[\rho^2 + (z - z')^2\right]^{1/2} \ , \tag{69}$$

the ρ and z components of the electric field at P due to a sinusoidal current filament of arbitrary phase,

$$I = \sin(kz' - \Theta_o) \ , \ z_1 < z' < z_2 \ , \tag{70}$$

are

$$E_\rho^f(\rho,z) = \frac{-j\eta}{2k^2\lambda\rho}\left[(z'-z)\ I\ \frac{\partial G_o}{\partial z'} + I\ G_o - (z'-z)\ G_o \frac{\partial I}{\partial z'}\right]_{z_1}^{z_2} \ , \tag{71}$$

$$E_z^f(\rho,z) = \frac{j\eta}{2k^2\lambda}\left[G_o \frac{\partial I}{\partial z'} - I\ \frac{\partial G_o}{\partial z'}\right]_{z_1}^{z_2} \ . \tag{72}$$

For a current that is constant over the length of the segment with strength I, the fields are

$$E_\rho^f(\rho,z) = \frac{I}{\lambda}\,\frac{j\eta}{2k^2}\ \left[\frac{\partial G_o}{\partial \rho}\right]_{z_1}^{z_2} \ , \tag{73}$$

$$E_z^f(\rho,z) = -\frac{I}{\lambda}\,\frac{j\eta}{2k^2}\left\{\left[\frac{\partial G_o}{\partial z'}\right]_{z_1}^{z_2} + k^2 \int_{z_1}^{z_2} G_o\ dz'\right\} \ . \tag{74}$$

These field expressions are exact for the specified currents. The integral over z' of G_o is evaluated numerically in NEC.

Substituting sine and cosine currents and evaluating the derivatives yields the following equations for the fields. For

$$I = I_o \begin{pmatrix} \sin kz' \\ \cos kz' \end{pmatrix} , \tag{75}$$

$$\begin{aligned} E_\rho^f(\rho,z) = \frac{-I_o}{\lambda} \frac{j\eta}{2k^2\rho} G_o \Bigg\{ & k(z-z') \begin{pmatrix} \cos kz' \\ -\sin kz' \end{pmatrix} \\ & + \left[1 - (z-z')^2 (1+jkr_o) \frac{1}{r_o^2} \right] \begin{pmatrix} \sin kz' \\ \cos kz' \end{pmatrix} \Bigg\} \Bigg|_{z_1}^{z_2} , \end{aligned} \tag{76}$$

$$\begin{aligned} E_z^f(\rho,z) = \frac{I_o}{\lambda} \frac{j\eta}{2k^2} G_o \Bigg\{ & k \begin{pmatrix} \cos kz' \\ -\sin kz' \end{pmatrix} \\ & - (1+jkr_o)(z-z') \frac{1}{r_o^2} \begin{pmatrix} \sin kz' \\ \cos kz' \end{pmatrix} \Bigg\} \Bigg|_{z_1}^{z_2} . \end{aligned} \tag{77}$$

For a constant current of strength I_o,

$$E_\rho^f(\rho,z) = -\frac{I_o}{\lambda} \frac{j\eta\rho}{2k^2} \left[(1+jkr_o) \frac{G_o}{r_o^2} \right]_{z_1}^{z_2} , \tag{78}$$

$$\begin{aligned} E_z^f(\rho,z) = -\frac{I_o}{\lambda} \frac{j\eta}{2k^2} \Bigg\{ & \left[(1+jkr_o)(z-z') \frac{G_o}{r_o^2} \right]_{z_1}^{z_2} \\ & + k^2 \int_{z_1}^{z_2} G_o \, dz' \Bigg\} . \end{aligned} \tag{79}$$

Despite the seemingly crude approximation, the thin-wire kernel does accurately represent the effect of wire radius for wires that are sufficiently thin. The accuracy range was studied by Poggio and Adams (ref. 11) where an

extended thin-wire kernel was developed for wires that are too thick for the thin-wire approximation.

The derivation of the extended thin-wire kernel starts with the current on the surface of the source segment with surface density,

$$J(z') = I(z')/(2\pi a) ,$$

where a is the radius of the source segment. The geometry for evaluation of the fields is shown in figure 5. A current filament of strength $Id\phi/(2\pi)$ is integrated over ϕ with

$$\rho' = [\rho^2 + a^2 - 2a\rho \cos \phi]^{1/2} , \tag{80}$$

$$r = [\rho'^2 + (z-z')^2]^{1/2} . \tag{81}$$

Thus, the z component of the field of the current tube is

$$E_z^t(\rho, z) = \frac{1}{2\pi} \int_0^{2\pi} E_z^f(\rho', z) d\phi . \tag{82}$$

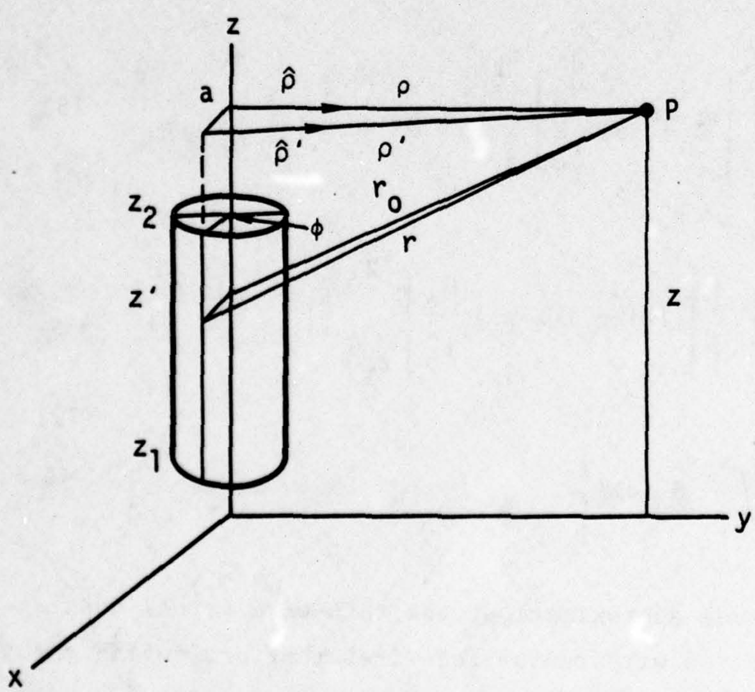


Figure 5. Current Geometry for the Extended Thin-Wire Kernel.

For the ρ component of field, the change in the direction of $\hat{\rho}'$ must be considered. The field in the direction $\hat{\rho}$ is

$$E_z^t(\rho,z) = \frac{1}{2\pi}\int_0^{2\pi} E_\rho^f(\rho',z)(\hat{\rho}\cdot\hat{\rho}')d\phi \ , \tag{83}$$

where

$$\hat{\rho}\cdot\hat{\rho}' = \frac{\rho - a\cos\phi}{\rho'} = \frac{\partial\rho'}{\partial\rho}$$

The integrals over ϕ in equations (82) and (83) cannot be evaluated in closed form. Poggio and Adams, however, have evaluated them as a series in powers of a^2 (ref. 11). The first term in the series gives the thin-wire kernel. For the extended thin-wire kernel, the second term involving a^2 is retained with terms of order a^4 neglected. As with the thin-wire kernel, the field observation point is on the segment surface. Hence, when evaluating the field on the source segment, $\rho = a$.

The field equations with the extended thin-wire approximation are given below. For a sinusoidal current of equation (70),

$$E_\rho(\rho,z) = \frac{-j\eta}{2k^2\lambda\rho}\left[(z'-z)I\frac{\partial G_2}{\partial z'} + IG_2 - (z'-z)G_2\frac{\partial I}{\partial z'}\right]_{z_1}^{z_2} , \tag{84}$$

$$E_z(\rho,z) = \frac{j\eta}{2k^2\lambda}\left[G_1\frac{\partial I}{\partial z'} - I\frac{\partial G_1}{\partial z'}\right]_{z_1}^{z_2} , \tag{85}$$

For a constant current of strength I_o,

$$E_\rho(\rho,z) = \frac{I}{\lambda}\frac{j\eta}{2k^2}\left[\frac{\partial G_1}{\partial\rho}\right]_{z_1}^{z_2} , \tag{86}$$

$$\begin{aligned} E_z(\rho,z) = -\frac{I}{\lambda}\frac{j\eta}{2k^2} & \left\{\left[\frac{\partial G_1}{\partial z'}\right]_{z_1}^{z_2}\right. \\ & \left. + k^2\left[1 - \frac{(ka)^2}{4}\right]\int_{z_1}^{z_2} G_o dz' - \frac{(ka)^2}{4}\left[\frac{\partial G_o}{\partial z'}\right]_{z_1}^{z_2}\right\}. \end{aligned} \tag{87}$$

The term G_1 is the series approximation of

$$G_1^t = \frac{1}{2\pi}\int_o^{2\pi} G\, d\phi , \tag{88}$$

where

$$G = \exp(-jkr)/r.$$

Neglecting terms of order a^4,

$$G_1 = G_o\left\{1 - \frac{a^2}{2r_o^2}(1+jkr_o) + \frac{a^2\rho^2}{4r_o^4}\left[3(1+jkr_o) - k^2r_o^2\right]\right\} , \tag{89}$$

$$\begin{aligned} \frac{\partial G_1}{\partial z'} = \frac{(z-z')}{r_o^2} G_o & \left\{(1+jkr_o) - \frac{a^2}{2r_o^2}\left[3(1+jkr_o) - k^2r_o^2\right]\right. \\ & \left. - \frac{a^2\rho^2}{4r_o^4}\left[jk^3r_o^3 + 6k^2r_o^2 - 15(1+jkr_o)\right]\right\} , \end{aligned} \tag{90}$$

$$\frac{\partial G_1}{\partial \rho} = -\frac{\rho G_o}{r_o^2}\left\{(1+jkr_o) - \frac{a^2}{r_o^2}\left[3(1+jkr_o) - k^2r_o^2\right]\right.$$

$$\left. - \frac{a^2\rho^2}{4r_o^4}\left[jk^3r_o^3 + 6k^2r_o^2 - 15(1+jkr_o)\right]\right\} . \tag{91}$$

The term G_2 is the series approximation of

$$G_2^t = \frac{1}{2\pi}\int_o^{2\pi} \frac{\rho - a\cos\phi}{\rho'^2} G \, d\phi . \tag{92}$$

To order a^2,

$$G_2 = \frac{G_o}{\rho}\left\{1 + \frac{a^2\rho^2}{4r_o^4}\left[3(1+jkr_o) - k^2r_o^2\right]\right\} , \tag{93}$$

$$\frac{\partial G_2}{\partial z'} = \frac{(z-z')}{\rho r_o^2} G_o \left\{(1+jkr_o) - \frac{a^2\rho^2}{4r_o^4}\left[jk^3r_o^3 + 6k^2r_o^2 - 15(1+jkr_o)\right]\right\} . \tag{94}$$

Equation (86) makes use of the relation

$$(\hat{\rho}\cdot\hat{\rho}')\frac{\partial G}{\partial \rho'} = \frac{\partial G}{\partial \rho'}\frac{\partial \rho'}{\partial \rho} = \frac{\partial G}{\partial \rho} , \tag{95}$$

while equation (87) follows from

$$G_1 = \left[1 - \frac{(ka)^2}{4} - \frac{a^2}{4}\frac{\partial^2}{\partial z'^2}\right] G_o . \tag{96}$$

When the observation point is within the wire ($\rho<a$), a series expansion in ρ rather than a is used for G_o and G_2. For G_1 this simply involves interchanging ρ and a in equations (89) and (90). Then for $\rho < a$, with

$$r_a = \left[a^2 + (z-z')^2\right]^{1/2} , \tag{97}$$

$$G_a = \exp(-jkr_a)/r_a \;, \tag{98}$$

the expressions for G_1, G_2 and their derivatives are

$$G_1 = G_a \left\{ 1 - \frac{\rho^2}{2r_a^2}(1+jkr_a) + \frac{a^2\rho^2}{4r_a^4}\left[3(1+jkr_a) - k^2r_a^2\right]\right\} \;, \tag{99}$$

$$\begin{aligned} \frac{\partial G_1}{\partial z'} = \frac{(z-z')}{r_a^2} G_a \Bigg\{ &(1+jkr_a) - \frac{\rho^2}{2r_a^2}\left[3(1+jkr_a) - k^2r_a^2\right] \\ &- \frac{a^2\rho^2}{4r_a^4}\left[jk^3r_a^3 + 6k^2r_a^2 - 15(1+jkr_a)\right]\Bigg\} \;, \end{aligned} \tag{100}$$

$$\frac{\partial G_1}{\partial \rho} = -\frac{\rho}{r_a^2} G_a \left\{(1+jkr_a) - \frac{a^2}{2r_a^2}\left[3(1+jkr_a) - k^2r_a^2\right]\right\} \;, \tag{101}$$

$$G_2 = -\frac{\rho}{2r_a^2} G_a \,(1+jkr_a) \;, \tag{102}$$

$$\frac{\partial G_2}{\partial z'} = -\frac{(z-z')\rho}{2r_a^4} G_a \left[3(1+jkr_a) - k^2r_a^2\right] \;. \tag{103}$$

Special treatment of bends in wires is required when the extended thin-wire kernel is used. The problem stems from the cancellation of terms evaluated at z_1 and z_2 in the field equations when segments are part of a continuous wire. The current expansion in NEC results in a current having a continuous value and derivative along a wire without junctions. This ensures that for two adjacent segments on a straight wire, the contributions to the z component of electric field at z_2 of the first segment exactly cancel the contributions from z_1, representing the same point, for the second segment. For a straight wire of several segments, the only contributions to E_z with either the thin-wire or extended thin-wire kernel come from the two wire ends and the integral of G_o along the wire. For the ρ component of field or either component at a bend, while there is not complete cancellation, there may be partial cancellation of large end contributions.

The cancellation of end terms makes necessary a consistent treatment of the current on both sides of a bend for accurate evaluation of the field. This is easily accomplished with the thin-wire kernel since the current filament on the wire axis is physically continuous around a bend. However, the current tube assumed for the extended thin-wire kernel cannot be continuous around its complete circumference at a bend. This was found to reduce the solution accuracy when the extended thin-wire kernel was used for bent wires.

To avoid this problem in NEC, the thin-wire form of the end terms in equations (71) through (74) is always used at a bend or change in radius. The extended thin-wire kernel is used only at segment ends where two parallel segments join, or at free ends. The switch from extended thin-wire form to the thin-wire form is made from one end of a segment to the other rather than between segments where the cancellation of terms is critical.

When segments are separated by a large distance, the interaction may be computed with sufficient accuracy by treating the segment current as an infinitesimal current element at the segment center. In spherical coordinates, with the segment at the origin along the $\Theta = 0$ axis, the electric field is

$$E_r(r,\Theta) = \frac{M\eta}{2\pi r^2} \exp(-jkr) \left(1 - \frac{j}{kr}\right) \cos\Theta ,$$

$$E_\Theta(r,\Theta) = \frac{M\eta}{4\pi r^2} \exp(-jkr) \left(1 + jkr - \frac{j}{kr}\right) \sin\Theta .$$

The dipole moment M for a constant current I on a segment of length Δ_i is

$$M = I\,\Delta_i .$$

For a current $I\cos[k(s - s_i)]$ with $|s - s_i| < \Delta_i/2$,

$$M = \frac{2I}{k} \sin(k\Delta_i/2) ,$$

while for a current $I\sin[k(s - s_i)]$,

$$M = 0 .$$

Use of this approximation saves a significant amount of time in evaluating the interaction matrix elements for large structures. The minimum interaction distance at which it is used is selected by the user in NEC. A default distance of one wavelength is set, however.

For each of the three methods of computing the field at a segment due to the current on another segment, the field is evaluated on the surface of the observation segment. Rather than choosing a fixed point on the segment surface, the field is evaluated at the cylindrical coordinates ρ', z with the source segment at the origin. If the center point on the axis of the observation segment is at ρ, z, then

$$\rho' = \left[\rho^2 + a_o^2\right]^{1/2} ,$$

where a_o is the radius of the observation segment. Also, the component of E_ρ tangent to the observation segment is computed as

$$\vec{E}_\rho \cdot \hat{s} = (\hat{\rho}\cdot\hat{s}) \frac{\rho}{\rho'} E_\rho .$$

Inclusion of the factor ρ/ρ', which is the cosine of the angle between $\hat{\rho}$ and $\hat{\rho}'$, is necessary for accurate results at bends in thick wires.

4. THE MATRIX EQUATION FOR CURRENT

For a structure having N_s wire segments and N_p patches, the order of the matrix in equation (19) is $N = N_s + 2N_p$. In NEC the wire segment equations occur first in the linear system so that, in terms of submatrices, the equation has the form

$$\begin{bmatrix} A & B \\ & \\ C & D \end{bmatrix} \begin{bmatrix} I_w \\ \\ I_p \end{bmatrix} = \begin{bmatrix} E_w \\ \\ H_p \end{bmatrix} ,$$

with equations derived from equation (14) in odd numbered rows in the lower set and equation (15) in even rows. I_w is then the column vector of segment

basis function amplitudes, and I_p is the patch-current amplitudes (J_{1j}, J_{2j}, $j=1,\ldots,N_p$). The elements of E_w are the left-hand side of equation (13) evaluated at segment centers, while H_p contains, alternately, the left-hand sides of equations (14) and (15) evaluated at patch centers.

A matrix element A_{ij} in submatrix A represents the electric field at the center of segment i due to the j^{th} segment basis function, centered on segment j. A matrix element D_{ij} in submatrix D represents a tangential magnetic field component at patch k due to a surface-current pulse on patch ℓ where

$$k = \mathrm{Int}\left[(i-1)/2\right] + 1 \; ,$$

$$\ell = \mathrm{Int}\left[(j-1)/2\right] + 1 \; ,$$

and Int[] indicates truncation. The source pulse is in the direction $\hat{t}_1$ when j is odd, and direction $\hat{t}_2$ when j is even. When $k = \ell$ the contribution of the surface integral is zero since the vector product is zero on the flat patch surface, although a ground image may produce a contribution. However, for $k = \ell$, there is a contribution of $\pm$ 1/2 from the coefficient of $\vec{J}_s(\vec{r})$ in equation (14) or (15). Matrix elements in submatrices B and C represent electric fields due to surface-current pulses and magnetic fields due to segment basis functions, respectively. These present no special problems since the source and observation points are always separated.

5. SOLUTION OF THE MATRIX EQUATION

The matrix equation,

$$[G]\,[I] = [E] \; , \tag{104}$$

is solved in NEC by Gauss elimination (ref. 19). The basic step is factorization of the matrix G into the product of an upper triangular matrix U and a lower triangle matrix L where

$$G = [L]\,[U] \; .$$

The matrix equation is then

$$[L]\,[U]\,[I] = [E] \; , \tag{105}$$

from which the solution, I, is computed in two steps as

$$[L]\ [F] = [E]\ , \tag{106}$$

and

$$[U]\ [I] = [F]\ . \tag{107}$$

Equation (106) is first solved for F by forward substitution, and equation (107) is then solved for I by backward substitution.

The major computational effort is factoring G into L and U. This takes approximately 1/3 N^3 multiplication steps for a matrix of order N compared to N^3 for inversion of G by the Gauss-Jordan method. Solution of equations (106) and (107), making use of the triangular properties of L and U, takes approximately as many multiplications as would be required for multiplication of G^{-1} by the column vector E. The factored matrices L and U are saved in NEC since the solution for induced current may be repeated for a number of different excitations. This, then, requires only the repeated solution of equations (106) and (107).

Computation of the elements of the matrix G and solution of the matrix equation are the two most time-consuming steps in computing the response of a structure, often accounting for over 90% of the computation time. This may be reduced substantially by making use of symmetries of the structure, either symmetry about a plane, or symmetry under rotation.

In rotational symmetry, a structure having M sectors is unchanged when rotated by any multiple of 360/M degrees. If the equations for all segments and patches in the first sector are numbered first and followed by successive sectors in the same order, the matrix equation can be expanded in submatrices in the form

$$\begin{bmatrix} A_1 & A_2 & A_3 & \cdots\cdots & A_{M-1} & A_M \\ A_M & A_1 & A_2 & \cdots\cdots & A_{M-2} & A_{M-1} \\ A_{M-1} & A_M & A_1 & & A_{M-3} & A_{M-2} \\ \cdot & & & & \cdot & \cdot \\ \cdot & & & & \cdot & \cdot \\ \cdot & & & & \cdot & \cdot \\ \cdot & & & & \cdot & \cdot \\ A_2 & A_3 & A_4 & & A_M & A_1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ I_M \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ E_M \end{bmatrix} \tag{108}$$

If there are N_c equations in each sector, E_i and I_i are N_c element column vectors of the excitations and currents in sector i. A_i is a submatrix of order N_c containing the interaction fields in sector 1 due to currents in sector i. Due to symmetry, this is the same as the fields in sector k due to currents in sector i + k, resulting in the repetition pattern shown. Thus only matrix elements in the first row of submatrices need be computed, reducing the time to fill the matrix by a factor of 1/M.

The time to solve the matrix equation can also be reduced by expanding the excitation subvectors in a discrete Fourier series as

$$E_i = \sum_{k=1}^{M} S_{ik} \mathcal{E}_k \qquad i=1,\ldots,M \; , \tag{109}$$

$$\mathcal{E}_i = \frac{1}{M} \sum_{k=1}^{M} S^*_{ik} E_k \qquad i=1,\ldots,M \; , \tag{110}$$

where

$$S_{ik} = \exp\left[j2\pi(i-1)(k-1)/M\right] \; , \tag{111}$$

$j=\sqrt{-1}$, and * indicates the conjugate of the complex number. Examining a component in the expansion,

$$E = \begin{bmatrix} S_{1k} \mathcal{E}_k \\ S_{2k} \mathcal{E}_k \\ \cdot \\ \cdot \\ \cdot \\ S_{Mk} \mathcal{E}_k \end{bmatrix} \; , \tag{112}$$

it is seen that the excitation differs from sector to sector only by a uniform phase shift. This excitation of a rotationally symmetric structure results in a solution having the same form as the excitation, i.e.,

$$I = \begin{bmatrix} S_{1k} \, \mathcal{I}_k \\ S_{2k} \, \mathcal{I}_k \\ \cdot \\ \cdot \\ \cdot \\ S_{Mk} \, \mathcal{I}_k \end{bmatrix} \quad . \tag{113}$$

It can be shown that this relation between solution and excitation holds for any matrix having the form of that in equation (108). Substituting these components of E and I into equation (108) yields the following matrix equation of order N_c relating $\mathcal{I}_k$ to $\mathcal{E}_k$:

$$\left[S_{1k} \, A_1 + S_{2k} \, A_2 + \ldots + S_{Mk} \, A_M\right] \left[\mathcal{I}_k\right] = S_{1k} \left[\mathcal{E}_k\right] \quad . \tag{114}$$

The solution for the total excitation is then obtained by an inverse transformation,

$$I_i = \sum_{k=1}^{M} S_{ik} \, \mathcal{I}_k \qquad i=1, \ldots, M \; . \tag{115}$$

The solution procedure, then, is first to compute the M submatrices A_i and Fourier-transform these to obtain

$$\mathcal{A}_i = \sum_{k=1}^{M} S_{ik} \, A_k \qquad i=1, \ldots, M \; . \tag{116}$$

The matrices $\mathcal{A}_i$, of order N_c, are then each factored into upper and lower triangular matrices by the Gauss elimination method. For each excitation vector, the transformed subvectors are then computed by equation (110) and the transformed current subvectors are obtained by solving the M equations,

$$[\mathcal{A}_i] \; [\mathcal{I}_i] = [\mathcal{E}_i] \; . \tag{117}$$

The total solution is then given by equation (115).

The same procedure can be used for structures that have planes of symmetry. The Fourier transform is then replaced by even and odd excitations about each symmetry plane. All equations remain the same with the exception

that the matrix S with elements S_{ij}, given by equation (111), is replaced by the following matrices:

For one plane of symmetry,

$$S = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} ;$$

For two orthogonal planes of symmetry,

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} ;$$

and for three orthogonal symmetry planes,

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} .$$

For either rotational or plane symmetry, the procedure requires factoring of M matrices of order N_c rather than one matrix of order MN_c. Each excitation then requires the solution of the M matrix equations. Since the time for factoring is approximately proportional to the cube of the matrix order and the time for solution is proportional to the square of the order, the symmetry results in a reduction of factor time by M^{-2} and in solution time by M^{-1}. The time to compute the transforms is generally small compared to the time for matrix operations since it is proportional to a lower power of N_c. Symmetry also reduces the number of locations required for matrix storage by M^{-1} since only the first row of submatrices need be stored. The transformed matrices, $\mathcal{A}_i$, can replace the matrices A_i as they are computed.

NEC includes a provision to generate and factor an interaction matrix and save the result on a file. A later run, using the file, may add to the structure and solve the complete model without unnecessary repetition of calculations. This procedure is called the Numerical Green's Function (NGF) option since the effect is as if the free space Green's function in NEC were replaced by the Green's function for the structure on the file. The NGF is particularly useful for a large structure, such as a ship, on which various antennas will be added or modified. It also permits taking advantage of partial symmetry since a NGF file may be written for the symmetric part of a structure, taking advantage of the symmetry to reduce computation time. Unsymmetric parts can then be added in a later run.

For the NGF solution the matrix is partitioned as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix},$$

where A is the interaction matrix for the initial structure, D is the matrix for the added structure, and B and C represent mutual interactions. The current is computed as

$$I_2 = \left[D - CA^{-1}B\right]^{-1} \left[E_2 - CA^{-1} E_1\right] ,$$

$$I_1 = A^{-1} E_1 - A^{-1} BI_2 ,$$

after the factored matrix A has been read from the NGF file along with other necessary data.

Electrical connections between the new structure and the old (NGF) structure require special treatment. If a new wire or patch connects to an old wire the current basis function for the old wire segment is changed by the modified condition at the junction. The old basis function is given zero amplitude by adding a new equation having all zeros except for a one in the column of the old basis function. A new column is added for the corrected basis function. When a new wire connects to an old patch the patch must be divided into four new patches to apply the connection condition of equation (51). Hence both the current basis function and match point for the old patch are replaced.

Section IV
Effect of a Ground Plane

In the integral equation formulation used in NEC, a ground plane changes the solution in three ways: (1) by modifying the current distribution through near-field interaction; (2) by changing the field illuminating the structure; and (3) by changing the reradiated field. Effects (2) and (3) are easily analyzed by plane-wave reflection as a direct ray and a ray reflected from the ground. The reradiated field is not a plane wave when it reflects from the ground, but, as can be seen from reciprocity, plane-wave reflection gives the correct far-zone field. Analysis of the near-field interaction effect is, however, much more difficult.

In Section II, the kernels of the integral equations are free-space Green's functions, representing the E or H field at a point $\vec{r}$ due to an infinitesimal electric current element at $\vec{r}'$. When a ground is present the free space Green's functions must be replaced by Green's functions for the ground problem. The solution for the fields of current elements in the presence of a ground plane was developed by Arnold Sommerfeld (ref. 20). While this solution has been used directly in integral-equation computer codes, excessive computation time greatly limits its use. Numerous approximations to the Sommerfeld solution have been developed that require less time for evaluation but all have limited applicability.

The NEC code has three options for grounds. The most accurate for lossy grounds uses the Sommerfeld solution for interaction distances less than one wavelength and an asymptotic expansion for larger distances. To keep the solution time reasonable, a grid of values of the Sommerfeld solution is generated and interpolation is used to find specific values. This method is presently implemented only for wires in NEC but could be extended to patches. The solution for a perfectly conducting ground is much simpler since the ground may be replaced by the image of the currents above it. The third option models a lossy ground by a modified image method using the Fresnel plane-wave reflection coefficients. While specular reflection does not accurately describe the behavior of near fields, the approximation has been found to provide useful results for structures that are not too near to the ground (refs. 21, 22). The attraction of this method is its simplicity and speed of computation which are the same as for the image method for perfect ground.

1. THE SOMMERFELD/NORTON METHOD

The Sommerfeld/Norton ground option in NEC originated with the code WFLLL2A (ref. 23) which uses numerical evaluation of the Sommerfeld integrals for ground fields when the interaction distance is small and uses Norton's asymptotic approximations (ref. 24) for larger distances. Since evaluation of the Sommerfeld integrals is very time consuming, a code, SOMINT, was developed (ref. 25) which uses bivariate interpolation in a table of precomputed Sommerfeld integral values to obtain the field values needed for integration over current distributions. This method greatly reduces the required computation time. NEC uses a similar interpolation method with modifications to allow wires closer to the air-ground interface and to further reduce computation time. Although the code WFLLL2A allows wires both above and below the interface, both NEC and SOMINT are presently restricted to wires on the free-space side. The method used in NEC to evaluate the field over ground is described below, and the numerical evaluation of the Sommerfeld integrals to fill the interpolation grid is discussed in Section IV-2.

The electric field above an air-ground interface due to an infinitesimal current element of strength $I\ell$ also above the interface, with parameters shown in figure 6, is given by the following expressions:

$$E_\rho^V = C_1 \frac{\partial^2}{\partial\rho\partial z}\left[G_{22} - G_{21} + k_1^2 V_{22}\right] , \tag{118}$$

$$E_z^V = C_1\left(\frac{\partial^2}{\partial z^2} + k_2^2\right)\left(G_{22} - G_{21} + k_1^2 V_{22}\right) , \tag{119}$$

$$E_\rho^H = C_1 \cos\phi\left[\frac{\partial^2}{\partial\rho^2}\left(G_{22} - G_{21} + k_2^2 V_{22}\right) + k_2^2\left(G_{22} - G_{21} + U_{22}\right)\right] , \tag{120}$$

$$E_\phi^H = -C_1 \sin\phi\left[\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(G_{22} - G_{21} + k_2^2 V_{22}\right) + k_2^2\left(G_{22} - G_{21} + U_{22}\right)\right] , \tag{121}$$

$$E_z^H = C_1 \cos\phi \frac{\partial^2}{\partial z\partial\rho}\left(G_{22} + G_{21} - k_1^2 V_{22}\right) , \tag{122}$$

$$C_1 = \frac{-j\omega I\ell\mu_o}{4\pi k_2^2} , \tag{123}$$

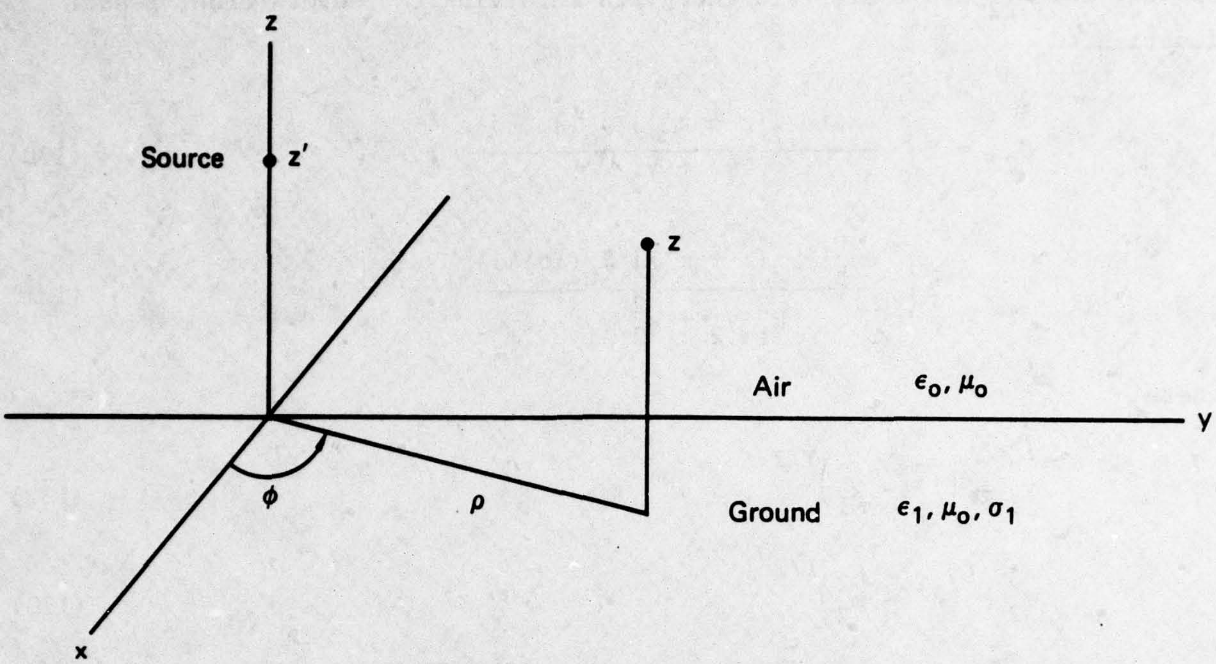


Figure 6. Coordinates for Evaluating the Field of a Current Element Over Ground.

$$k_1^2 = \omega^2 \mu_o \varepsilon_o \left(\frac{\varepsilon_1}{\varepsilon_o} - \frac{j\sigma_1}{\omega\varepsilon_o}\right) , \tag{124}$$

$$k_2^2 = \omega^2 \mu_o \varepsilon_o , \tag{125}$$

where the superscript indicates a vertical (V) or horizontal (H) current element and the subscript indicates the cylindrical component of the field vector. The horizontal current element is along the x axis.

G_{22} and G_{21} are the free space and image Green's functions

$$G_{22} = \exp(-jk_2R_2)/R_2 , \tag{126}$$

$$G_{21} = \exp(-jk_2R_1)/R_1 , \tag{127}$$

where

$$R_1 = \left[\rho^2 + (z + z')^2\right]^{1/2} , \tag{128}$$

$$R_2 = \left[\rho^2 + (z - z')^2\right]^{1/2} , \tag{129}$$

and U_{22} and V_{22} are Sommerfeld integrals involving the zeroth order Bessel function, J_o

$$U_{22} = 2\int_o^\infty \frac{\exp[-\gamma_2(z + z')]\ J_o(\lambda\rho)\lambda d\lambda}{\gamma_1 + \gamma_2}\ , \tag{130}$$

$$V_{22} = 2\int_o^\infty \frac{\exp[-\gamma_2(z + z')]\ J_o(\lambda\rho)\lambda d\lambda}{k_1^2\ \gamma_2 + k_2^2\ \gamma_1}\ , \tag{131}$$

where

$$\gamma_1 = \left(\lambda^2 - k_1^2\right)^{1/2}\ , \tag{132}$$

$$\gamma_2 = \left(\lambda^2 - k_2^2\right)^{1/2}\ . \tag{133}$$

In NEC we need to compute the fields due to current filaments with arbitrary length and orientation by combining the field components in equations (118) through (122) and integrating over current distributions composed of constant, sine, and cosine components. Direct numerical integration over the segments is difficult due to singularities in the fields. G_{22} has a $1/R_2$ singularity while G_{21}, U_{22} and V_{22} each have $1/R_1$ singularities. The derivatives in the field expressions result in $1/R_3$ singularities with a triplet-like behavior in the field components parallel to the current filament. The resulting cancellation makes accurate numerical integration near the singularity very difficult.

The free-space field has a similar singularity, but as discussed in Section III-3, the integral over a straight filament may be evaluated in closed form for a sinusoidal current with free-space wavelength and involves only a numerical integration of G_{22} for a constant current. The dominant singular component of the ground field may be integrated in the same way. The terms involving G_{22} in equation (118) through (122) are, in fact, the field of the current element in free space, and their integral is obtained from the free-space routines in NEC.

The remaining terms represent the field due to ground and are singular at $R_1 = 0$. The singularities in U_{22} and V_{22} result from the failure of the integrals in equations (130) and (131) to converge without the exponential and

Bessel functions as ρ and $z + z'$ go to zero. The singular behavior of U_{22} and V_{22} as ρ and $z + z'$ go to zero may be found by setting $\gamma_1 = \gamma_2 = \lambda$ since the dominant contributions to the integrals for small ρ and $z + z'$ come from λ much greater than k_1 or k_2. Here, however, we only replace γ_1 by γ_2 and use the integrals

$$V_{22} \approx 2\int_o^\infty \frac{\exp[-\gamma_2(z + z')]\ J_o(\lambda\rho)\lambda d\lambda}{\gamma_2(k_1^2 + k_2^2)} = \frac{2G_{21}}{k_1^2 + k_2^2}\ , \tag{134}$$

$$U_{22} \approx \int_o^\infty \frac{\exp[-\gamma_2(z + z')]\ J_o(\lambda\rho)\lambda d\lambda}{\gamma_2} = G_{21} \tag{135}$$

$$|k_1|\rho << 1, \quad |k_1|(z + z') << 1,$$

which have the correct singular behavior and can be combined with the G_{21} terms. The field components due to ground [equation (108) through (122) without the G_{22} terms] may then be written as

$$G_\rho^V = C_1 \frac{\partial^2}{\partial\rho\partial z} k_1^2\, V_{22}' + C_1 \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2} \frac{\partial^2}{\partial\rho\partial z} G_{21}\ , \tag{136}$$

$$G_z^V = C_1\left(\frac{\partial^2}{\partial z^2} + k_2^2\right) k_1^2\, V_{22}' + C_1 \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2}\left(\frac{\partial^2}{\partial z^2} + k_2^2\right) G_{21} \tag{137}$$

$$\begin{aligned} G_\rho^H = {} & C_1 \cos\phi \left(\frac{\partial^2}{\partial\rho^2} k_2^2\, V_{22}' + k_2^2\, U_{22}'\right) \\ & - C_1 \cos\phi\, \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2}\left(\frac{\partial^2}{\partial\rho^2} + k_2^2\right) G_{21}\ , \end{aligned} \tag{138}$$

$$\begin{aligned} G_\phi^H = {} & -C_1 \sin\phi \left(\frac{1}{\rho}\frac{\partial}{\partial\rho} k_2^2\, V_{22}' + k_2^2\, U_{22}'\right) \\ & + C_1 \sin\phi\, \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2}\left(\frac{1}{\rho}\frac{\partial}{\partial\rho} + k_2^2\right) G_{21}\ , \end{aligned} \tag{139}$$

$$G_z^H = -\cos\phi \; G_\rho^V , \tag{140}$$

where

$$U'_{22} = U_{22} - \frac{2k_2^2}{k_2^2 + k_1^2} \; G_{21} ,$$

$$= 2\int_0^\infty \left[\frac{1}{\gamma_1 + \gamma_2} - \frac{k_2^2}{\gamma_2 (k_1^2 + k_2^2)} \right] \exp[-\gamma_2 (z + z')] \times J_0(\lambda\rho)\lambda d\lambda , \tag{141}$$

$$V'_{22} = V_{22} - \frac{2}{k_1^2 + k_2^2} \; G_{21} ,$$

$$= 2\int_0^\infty \left[\frac{1}{k_1^2 \, \gamma_2 + k_2^2 \, \gamma_1} - \frac{1}{\gamma_2 \; k_1^2 + k_2^2} \right] \times \exp[-\gamma_2 (z + z')] \; J_0(\lambda\rho)\lambda d\lambda . \tag{142}$$

In equations (136) through (140) the dominant singular component has been subtracted out of V_{22} and combined with G_{21}. The integral for V'_{22} converges without the exponential or Bessel function factors and remains finite as ρ and $z + z'$ go to zero. The derivatives of V'_{22} in the field expressions have $1/R_1$ singularities, but this is much less of a problem for numerical integration than the previous $1/R_1^3$ singularity. The singularity could be taken out of U_{22} also, but, instead, a term is taken out that results in the final terms in equations (136) through (139) being the image field multiplied by $(k_1^2 - k_2^2)/(k_1^2 + k_2^2)$. The integral over the current filament of these image terms is evaluated by the free-space equations leaving only the U'_{22} and V'_{22} terms to be integrated numerically. U'_{22} still has a $1/R_1$ singularity, but that is no worse than the derivatives of V'_{22}. With the thin-wire approximation, R_1 is never less than the wire radius so the integration is not difficult in practical cases.

The components left for numerical integration over the current distribution are then

$$F_\rho^V = C_1 \frac{\partial^2}{\partial\rho\partial z} k_1^2 V_{22}' , \tag{143}$$

$$F_z^V = C_1\left(\frac{\partial^2}{\partial z^2} + k_2^2\right) k_1^2 V_{22}' , \tag{144}$$

$$F_\rho^H = C_1 \cos\phi\left(\frac{\partial^2}{\partial\rho^2} k_2^2 V_{22}' + k_2^2 U_{22}'\right) , \tag{145}$$

$$F_\phi^H = -C_1 \sin\phi\left(\frac{1}{\rho}\frac{\partial}{\partial\rho} k_2^2 V_{22}' + k_2^2 U_{22}'\right) , \tag{146}$$

$$F_z^H = -\cos\phi\, F_\rho^V . \tag{147}$$

Since the integrals in equations (141) and (142) cannot be evaluated in closed form the following terms must be evaluated by numerical integration over λ:

$$\frac{\partial^2 V_{22}'}{\partial\rho^2} = \int_0^\infty D_2 \exp\left[-\gamma_2(z + z')\right] J_0''(\lambda\rho)\lambda^3 d\lambda , \tag{148}$$

$$\frac{\partial^2 V_{22}'}{\partial z^2} = \int_0^\infty D_2\, \gamma_2^2 \exp\left[-\gamma_2(z + z')\right] J_0(\lambda\rho)\lambda d\lambda , \tag{149}$$

$$\frac{\partial^2 V_{22}'}{\partial\rho\partial z} = -\int_0^\infty D_2\gamma_2 \exp\left[-\gamma_2(z + z')\right] J_0'(\lambda\rho)\lambda^2 d\lambda , \tag{150}$$

$$\frac{1}{\rho}\frac{\partial V_{22}'}{\partial\rho} = \frac{1}{\rho}\int_0^\infty D_2 \exp\left[-\gamma_2(z + z')\right] J_0'(\lambda\rho)\lambda^2 d\lambda , \tag{151}$$

$$V_{22}' = \int_0^\infty D_2 \exp\left[-\gamma_2(z + z')\right] J_0(\lambda\rho)\lambda d\lambda , \tag{152}$$

$$U_{22}' = \int_0^\infty D_1 \exp\left[-\gamma_2(z + z')\right] J_0(\lambda\rho)\lambda d\lambda , \tag{153}$$

where

$$D_1 = \frac{2}{\gamma_2 + \gamma_2} - \frac{2\, k_2^2}{\gamma_2 (k_1^2 + k_2^2)} \ , \tag{154}$$

$$D_2 = \frac{2}{k_1^2\, \gamma_2 + k_2^2\, \gamma_1} - \frac{2}{\gamma_2 (k_1^2 + k_2^2)} \ . \tag{155}$$

Evaluating these integrals over λ for each point needed in the numerical integration over the current distribution is slow on even the fåstest computers. Hence an interpolation technique is used for the remaining field components as was done in the code SOMINT for the total field due to ground. Since the integrals depend only on ρ and $z + z'$ a grid of values is generated for the field components of equations (143) through (146) and bivariate interpolation is used to obtain values for integration over a current distribution.

To facilitate interpolation in the region of the $1/R_1$ singularity, the components are divided by a function having a similar singularity and interpolation is performed on the ratio. The field components of equations (143) through (146) are divided by $\exp(-jkR_1)/R_1$ for all values of R_1 to remove the singularity and the free-space phase factor before interpolation. The factors $\sin\phi$ or $\cos\phi$ are also omitted until after interpolation to avoid introducing the ϕ dependence. The surfaces to which interpolation is applied are then

$$I_\rho^V = C_1\, R_1\, \exp(jkR_1)\, \frac{\partial^2}{\partial\rho\partial z}\, k_1^2\, V'_{22} \ , \tag{156}$$

$$I_z^V = C_1\, R_1\, \exp(jkR_1)\left(\frac{\partial^2}{\partial z^2} + k_2^2\right) k_1^2\, V'_{22} \ , \tag{157}$$

$$I_\rho^H = C_1\, R_1\, \exp(jkR_1)\left(\frac{\partial^2}{\partial\rho^2}\, k_2^2\, V'_{22} + k_2^2\, U'_{22}\right) \ , \tag{158}$$

$$I_\phi^H = -\, C_1\, R_1\, \exp(jkR_1)\left(\frac{1}{\rho}\frac{\partial}{\partial\rho}\, k_2^2\, V'_{22} + k_2^2\, U'_{22}\right) \ . \tag{159}$$

After interpolation on the smoothed surfaces the results are multiplied by the omitted factors to give the correct values.

With the singularity removed, interpolation may be used for arbitrarily small values of ρ and $z + z'$. The values for $R_1 = 0$ in the interpolation grid must be found as limits for R_1 approaching zero, however, since the integrals

do not converge in this case. When ρ and $z + z'$ approach zero the dominant contributions in equations (148) through (153) come from large λ. Hence the singular behavior can be found by setting γ_1 and γ_2 equal to λ. First, however, it is necessary to approximate D_1 and D_2 for $|\lambda| >> |k_1|$ as

$$D_1 = C_2/\lambda \; , \; C_2 = \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2} \; , \tag{160}$$

$$D_2 = C_3/\lambda^3, \; C_3 = \frac{k_2^2\left(k_1^2 - k_2^2\right)}{\left(k_1^2 + k_2^2\right)^2} \; . \tag{161}$$

For $|k_1|\rho << 1$ and $|k_1|(z + z') << 1$ the integrals become

$$\begin{aligned} \frac{\partial^2 V'_{22}}{\partial \rho^2} &\approx C_3 \int_0^\infty \exp\left[-\lambda(z + z')\right] J''_0 (\lambda\rho) \, d\lambda \; , \\ &= C_3\left[\frac{1 - \sin\theta}{\cos^2\theta} - 1\right] \frac{1}{R_1} \; , \end{aligned} \tag{162}$$

$$\frac{\partial^2 V'_{22}}{\partial z^2} \approx C_3 \int_0^\infty \exp\left[-\lambda(z + z')\right] J_0 (\lambda\rho) d\lambda = \frac{C_3}{R_1} \; , \tag{163}$$

$$\frac{\partial^2 V'_{22}}{\partial\rho\partial z} \approx - C_3 \int_0^\infty \exp\left[-\lambda(z + z')\right] J'_0(\lambda\rho) d\lambda = \frac{C_3(1 - \sin\theta)}{R_1 \cos\theta} \; , \tag{164}$$

$$\frac{1}{\rho} \frac{\partial V'_{22}}{\partial \rho} \approx \frac{C_3}{\rho} \int_0^\infty \exp\left[-\lambda(z + z')\right] J'_0 (\lambda\rho) \frac{1}{\lambda} \, d\lambda = \frac{-C_3(1 - \sin\theta)}{R_1 \cos^2\theta} \; , \tag{165}$$

$$U'_{22} \approx C_2 \int_0^\infty \exp\left[-\lambda(z + z')\right] J_0 (\lambda\rho) \, d\lambda = \frac{C_2}{R_1} \; , \tag{166}$$

where

$$R_1 = \left[\rho^2 + (z + z')^2\right]^{1/2} \; , \tag{167}$$

$$\theta = \tan^{-1}\left[(z + z')/\rho\right] \; . \tag{168}$$

V'_{22} remains finite as R_1 goes to zero and hence is neglected. Equations (156) through (159) for R_1 approaching zero are then

$$I_\rho^V = C_1 C_3 k_1^2 \left(\frac{1 - \sin\theta}{\cos\theta}\right) , \tag{169}$$

$$I_z^V = C_1 C_3 k_1^2 , \tag{170}$$

$$I_\rho^H = C_1 k_2^2 \left[C_2 - C_3 + C_3\left(\frac{1 - \sin\theta}{\cos^2\theta}\right)\right] , \tag{171}$$

$$I_\phi^H = C_1 k_2^2 \left[C_2 - C_3\left(\frac{1 - \sin\theta}{\cos^2\theta}\right)\right] . \tag{172}$$

Since the limiting values as R_1 goes to zero are functions of θ it is necessary to use R_1 and θ as the interpolation variables rather than ρ and $z + z'$.

Figures 7 through 10 are plots of the surfaces to which interpolation is applied for typical ground parameters. The width of the region of relatively rapid variation along the R_1 axis appears to be proportional to the wavelength

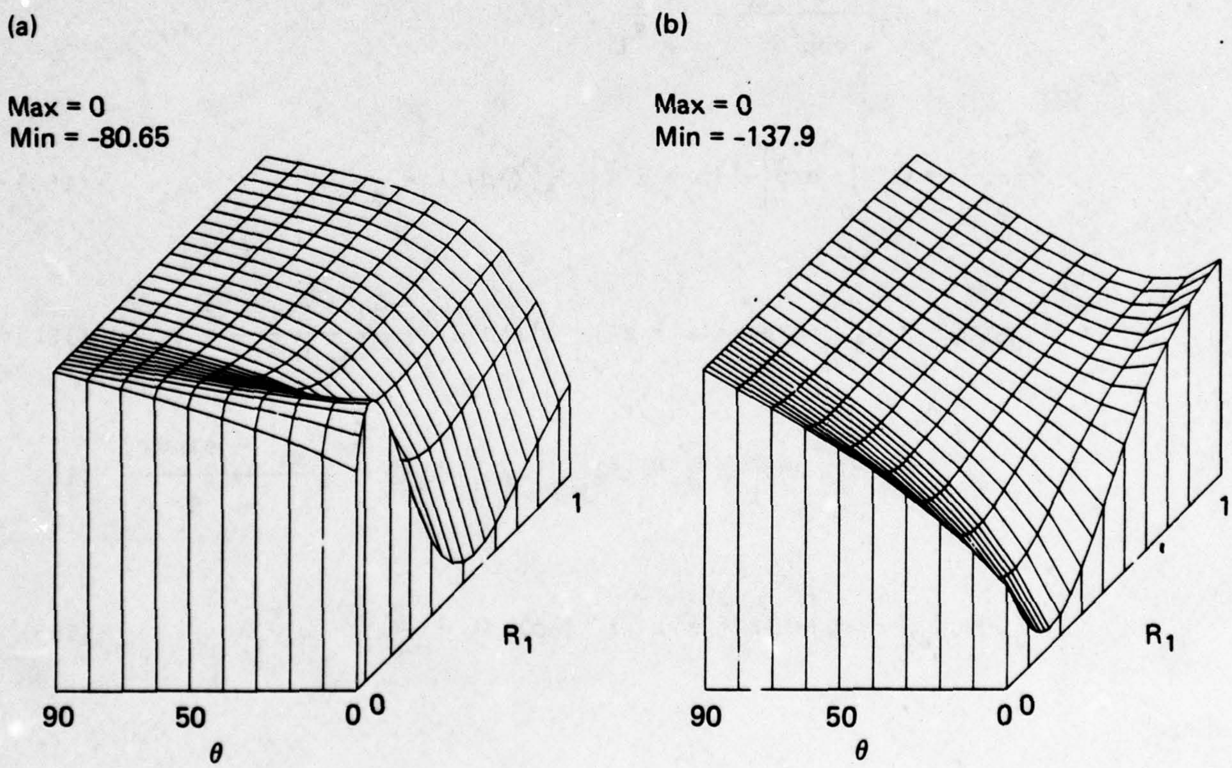


Figure 7. Real (a) and Imaginary (b) Parts of I_ρ^V for $\varepsilon_1/\varepsilon_o = 4$, $\sigma_1 = 0.001$ mhos/m, frequency = 10 MHz.

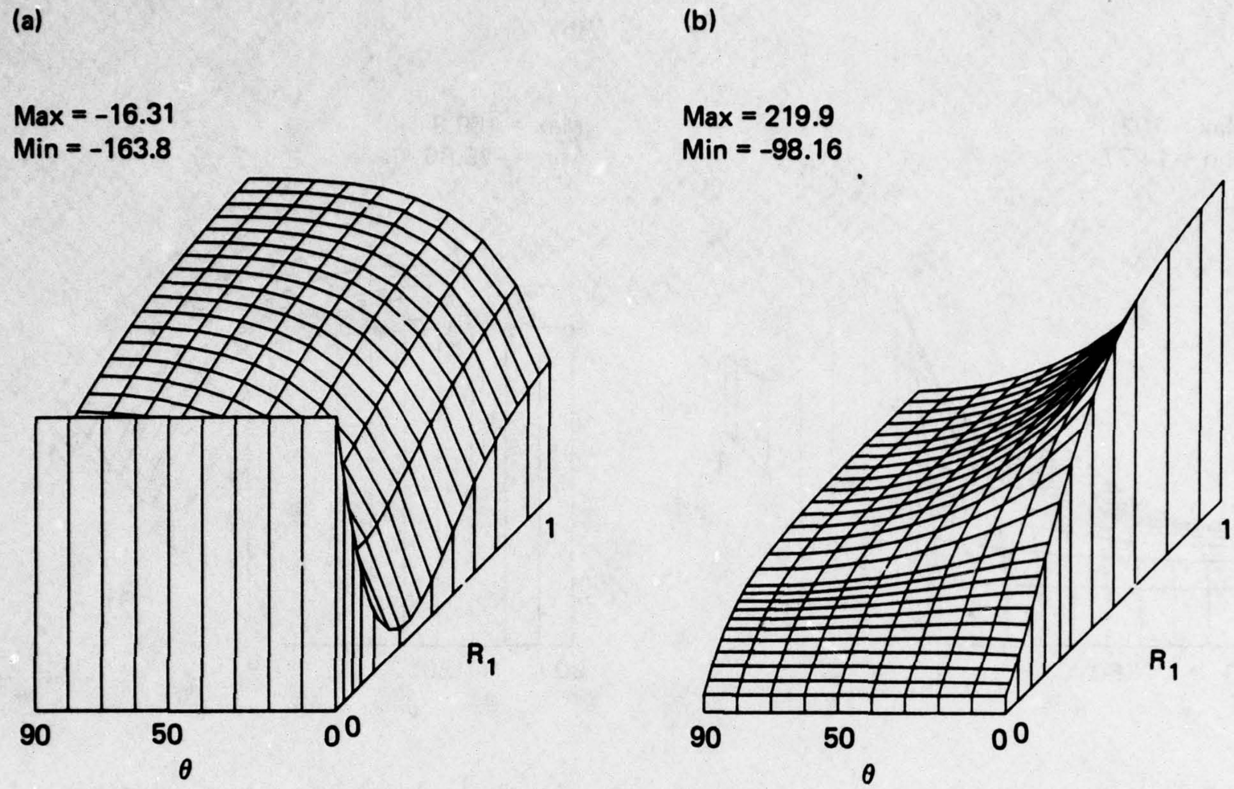


Figure 8. Real (a) and Imaginary (b) Parts of I_z^V for $\varepsilon_1/\varepsilon_o = 4$, $\sigma_1 = 0.001$ mhos/m, frequency = 10 MHz.

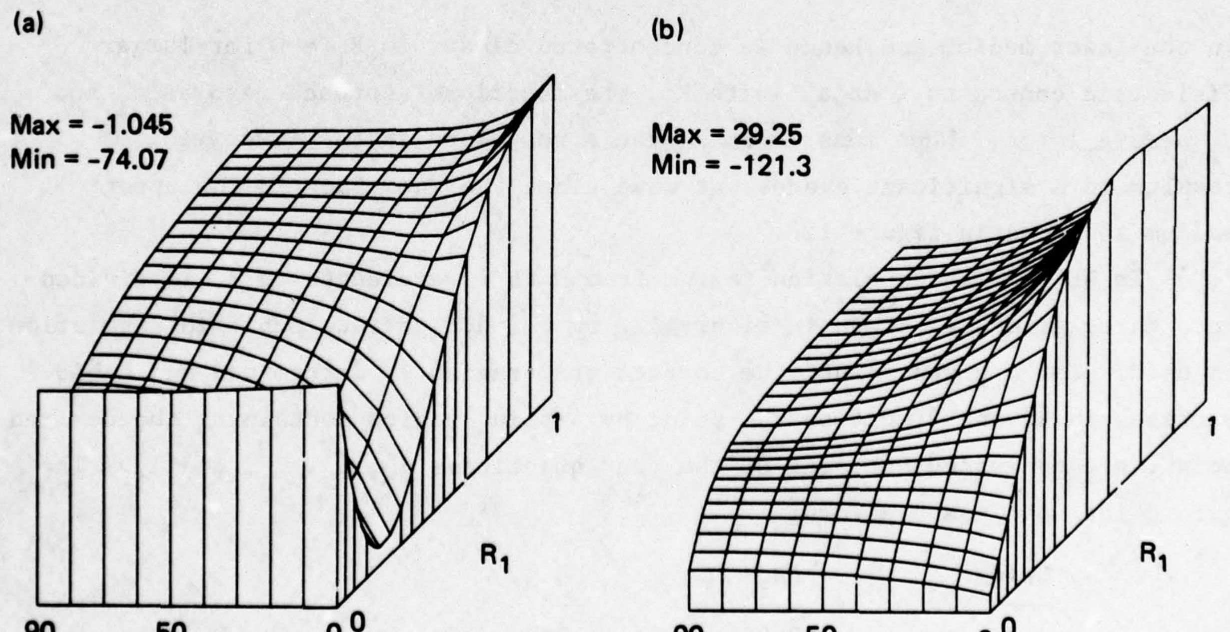
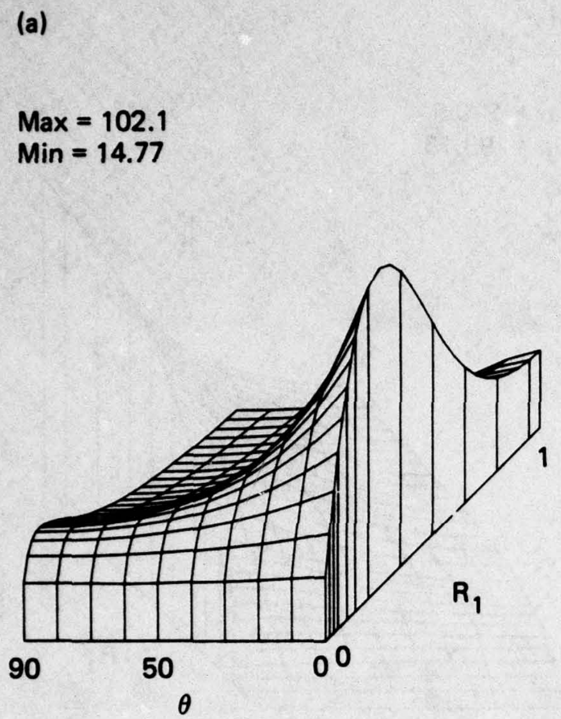


Figure 9. Real (a) and Imaginary (b) Parts of I_ρ^H for $\varepsilon_1/\varepsilon_o = 4$, $\sigma_1 = 0.001$ mhos/m, frequency = 10 MHz.



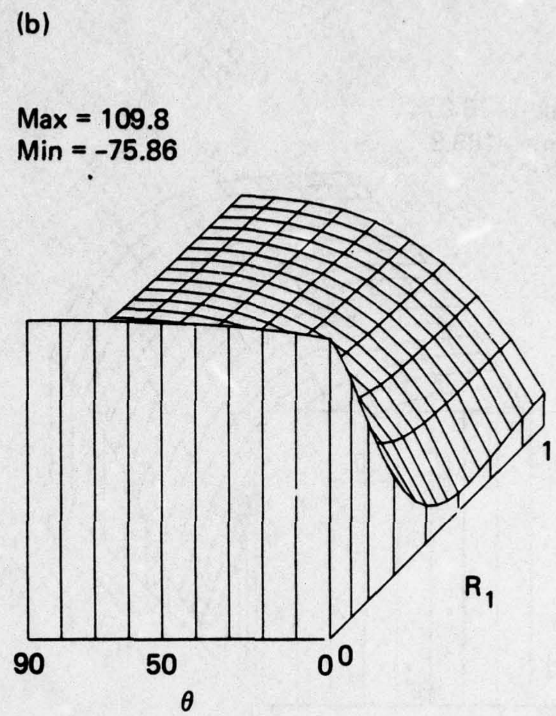


Figure 10. Real (a) and Imaginary (b) Parts of I_ϕ^H for $\varepsilon_1/\varepsilon_o = 4$, $\sigma_1 = 0.001$ mhos/m, frequency = 10 MHz.

in the lower medium and hence is concentrated closer to $R_1 = 0$ for larger dielectric constants. At a finite R_1, the functions approach zero as ε_1 and σ_1 become large. When loss is small the strong wave in the lower medium results in a significant evanescent wave along the interface in the upper medium as shown in figure 11.

In NEC the interpolation region from 0 to 1 wavelength in R_1 is divided into three grids, as shown in figure 12, on which bivariate cubic interpolation is used. For a given point, the correct grid region is determined and cubic surfaces in R_1 and θ, fit to a 4-point by 4-point region containing the desired point, are evaluated for each of the four quantities I_ρ^V, I_z^V, I_ρ^H, and I_ϕ^H. The grid point spacings used are:

Grid	ΔR_1	$\Delta\theta$
1	0.02λ	10°
2	0.05λ	5°
3	0.1λ	10°

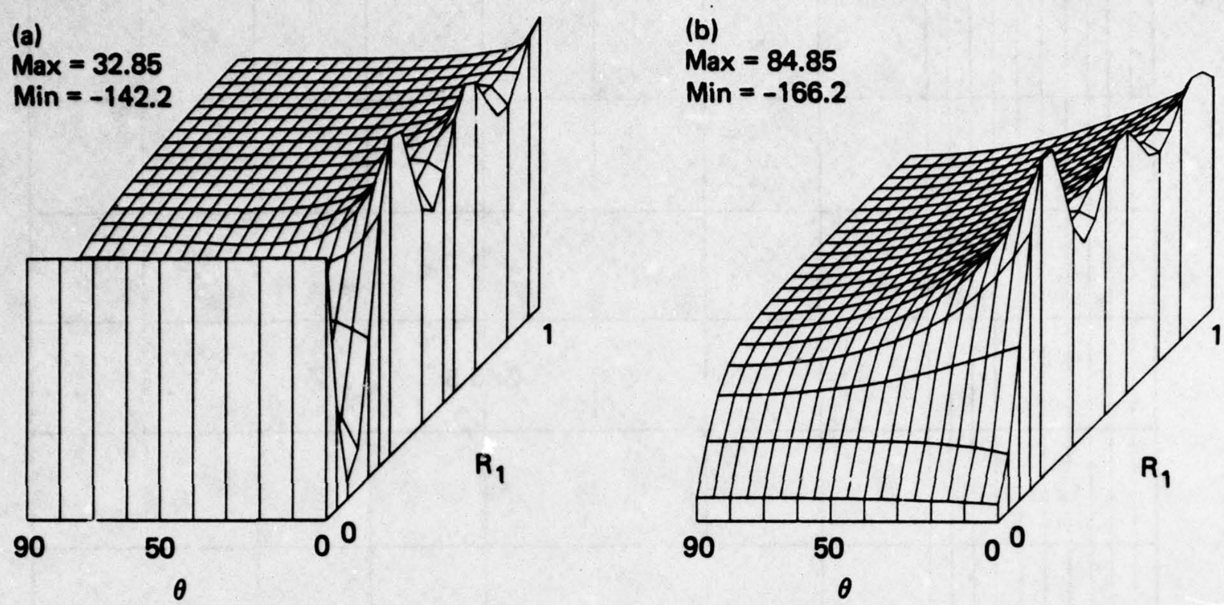


Figure 11. Real (a) and Imaginary (b) Parts of I_ρ^H for $\varepsilon_1/\varepsilon_o = 16$, $\sigma_1 = 0$.

These were determined by numerical tests to keep relative errors of interpolation generally in the range of 10^{-3} to 10^{-4}. A smaller ΔR_1 could be needed in grid 2 for large ε_1 and small σ_1 to handle the rapidly oscillating evanescent wave, but this is easily changed in the code.

The field evaluation in NEC uses variable-interval-width Romberg integration over the current distribution. At each integrand evaluation, the components I_ρ^V, I_z^V, I_ρ^H, and I_ϕ^H are obtained by interpolation, and the field components are combined according to the direction of the current. The numerical integral is then combined with the free-space field and with the image field multiplied by $(k_1^2 - k_2^2)/(k_1^2 + k_2^2)$ to obtain the total field over ground.

When R_1 from the observation point to the center of a wire segment is greater than one wavelength, the field is evaluated by Norton's asymptotic approximations (ref. 26) rather than the above method. Norton's formulas are given in Part II of this manual under subroutine GWAVE. Although they are less accurate than the Sommerfeld integral forms and require longer to evaluate than the interpolation, their use permits truncating the interpolation tables. Another approximation used for R_1 greater than a wavelength is to treat the current distribution on a segment as a lumped current element with the correct moment rather than integrating over the current distribution.

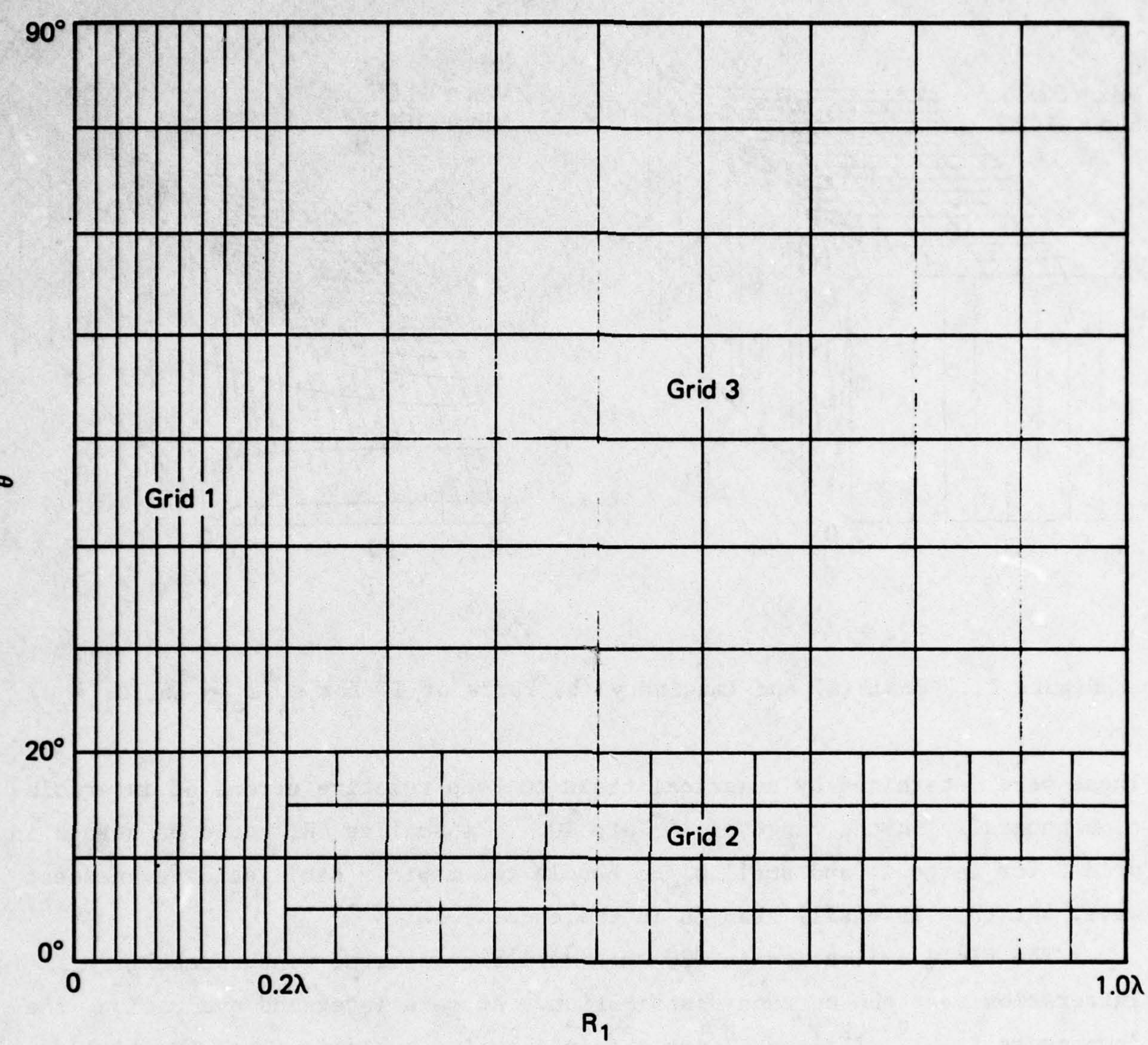


Fig. 12. Grid for Bivariate Interpolation of I's.

2. NUMERICAL EVALUATION OF THE SOMMERFELD INTEGRALS

The integrals in equations (148) through (153) are evaluated by numerical integration along contours in the complex λ plane. Although these integrals differ from the usual Sommerfeld integrals in the D_1 and D_2 terms, they are the same in the properties important to numerical integration — the locations of poles and branch cuts and the exponential behavior of the Bessel and exponential functions. The behavior of the integrands and numerical methods for evaluating the integrals are discussed in detail by Lytle and Lager

(ref. 27). This section describes the particular method used in NEC, which is basically the same as in the code WFLLL2A.

Since the integrands of the six integrals are similar, V'_{22} will be considered as typical. The integrands have branch cuts from $\pm k_1$ to infinity and $\pm k_2$ to infinity due to the square roots in γ_1 and γ_2 respectively. The branch cuts are chosen to be vertical, as shown in figure 13. The implications of this choice of branch cuts and the choice of Reimann sheets are discussed in ref. 27.

The key to rapid convergence in the numerical integration is to exploit the exponential behavior of the exponential and Bessel functions for large λ. The integration contour is deformed from the real axis into the complex plane, avoiding branch cuts and taking account of poles, to optimize convergence. With the vertical branch cuts chosen, there are no real poles on the primary Reimann sheet although virtual poles from D_1 or D_2 result in a near singularity in the region of $\pm k_2$ when k_1 approaches k_2 (free-space limit). Hence the integration contour should avoid the real axis in this region.

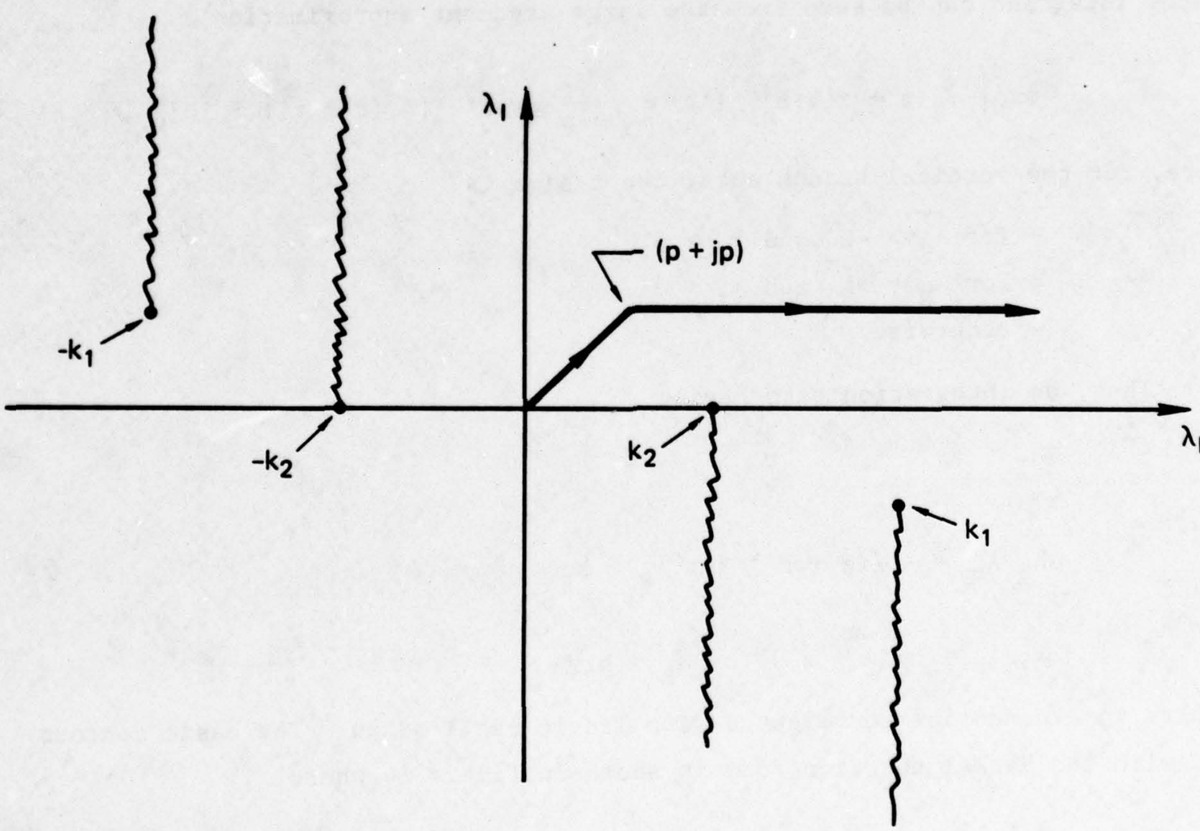


Figure 13. Contour for Evaluation of Bessel Function Form of Sommerfeld Integrals.

The contour used with the form of the integrals in equations (148) through (153) is shown in figure 13. The dominant factor for convergence in this case is the exponential function as λ_R increases. The Bessel function oscillates with slow convergence for increasing λ_R and grows exponentially as $|\lambda_I|$ increases. Hence it is of little help in convergence but restricts the contour to small $\rho|\lambda_I|$. The break in the contour is at $\lambda = p + jp$ where p is the minimum of $1/\rho$ and $1/(z + z')$.

Integration along this contour becomes difficult when $(z + z')/\rho$ is small since there may be many oscillations of the Bessel function before convergence. In this case an alternate form of the integrals is used which for V'_{22} is

$$V'_{22} = \frac{1}{2}\int_{-\infty}^{\infty} D_2 \exp\left[-\gamma_2(z + z')\right] H_o^{(2)} (\lambda\rho)\lambda d\lambda \ . \tag{173}$$

Since the Hankel function of type 2 decays exponentially as λ_I becomes negative, it provides rapid convergence without the exp $-\gamma_2(z + z')$ factor. The behavior of the integrand can be seen from the large argument approximation

$$\exp\left[-\gamma_2(z + z')\right] H_o^{(2)}(\lambda\rho) \approx \sqrt{\frac{2j}{\pi\lambda\rho}} \left\{\exp -\lambda[\pm\ (z + z') + j\rho]\right\} ,$$

where, for the vertical branch cuts, the $\pm$ sign is

$+$ for $\lambda_R > -k_2$ and $\lambda_I > 0$,
$+$ for $\lambda_R > k_2$ and $\lambda_I < 0$,
$-$ otherwise.

Thus, an integration path having

$$\lambda_I < 0$$

and

$$\lambda_I/\lambda_R = -\rho/(z + z') \text{ for } \lambda_R > k_2$$

or

$$\lambda_I/\lambda_R = \rho/(z + z') \text{ for } \lambda_R < k_2$$

results in exponential convergence with little oscillation. The basic contour used with the Hankel function form is shown in figure 14 where

$a = -j\ 0.4\ k_2$,
$b = (0.6 + j\ 0.2)k_2$,

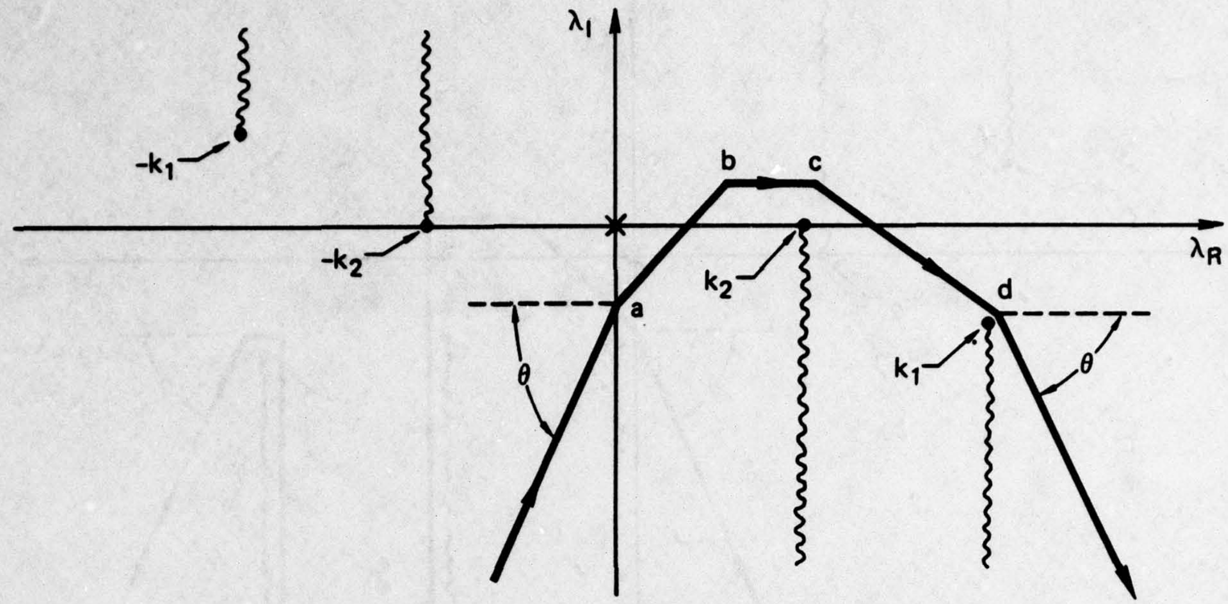


Figure 14. Contour for Evaluation of Hankel Function Form of Sommerfeld Integrals.

$$c = (1.02 + j\ 0.2)k_2\ ,$$
$$d = 1.01\ k_{1_R} + j\ 0.99\ k_{1_I}\ ;$$
$$\theta = \tan^{-1}\left(\frac{\rho}{z + z'}\right)\ .$$

To avoid the near singularity as k_1 approaches k_2, the real part of d is not allowed to be less than 1.1 k_2. This contour provides rapid convergence except when z + z' is small, $|k_1\rho|$ is large, and $k1_I/k1_R$ is small. There may then be many oscillations between c and d with little convergence. In such a case the contour in figure 15 is used where

$$e = k_1 + (-0.1 + j\ 0.2)\ ,$$
$$f = k_1 + (0.1 + j\ 0.2)\ .$$

The Hankel function form of the integrals provides rapid convergence for small z + z' including the case of z = z' = 0. For small ρ, however, the pole at $\lambda\rho = 0$ requires special treatment. In NEC the Hankel function form with the contour of figure 14 or 15 is used when ρ is greater than (z + z')/2 and the Bessel function form is used otherwise.

Integration along the contours is accomplished by adaptive interval-width Romberg integration. On the sections going to infinity, adaptive Romberg

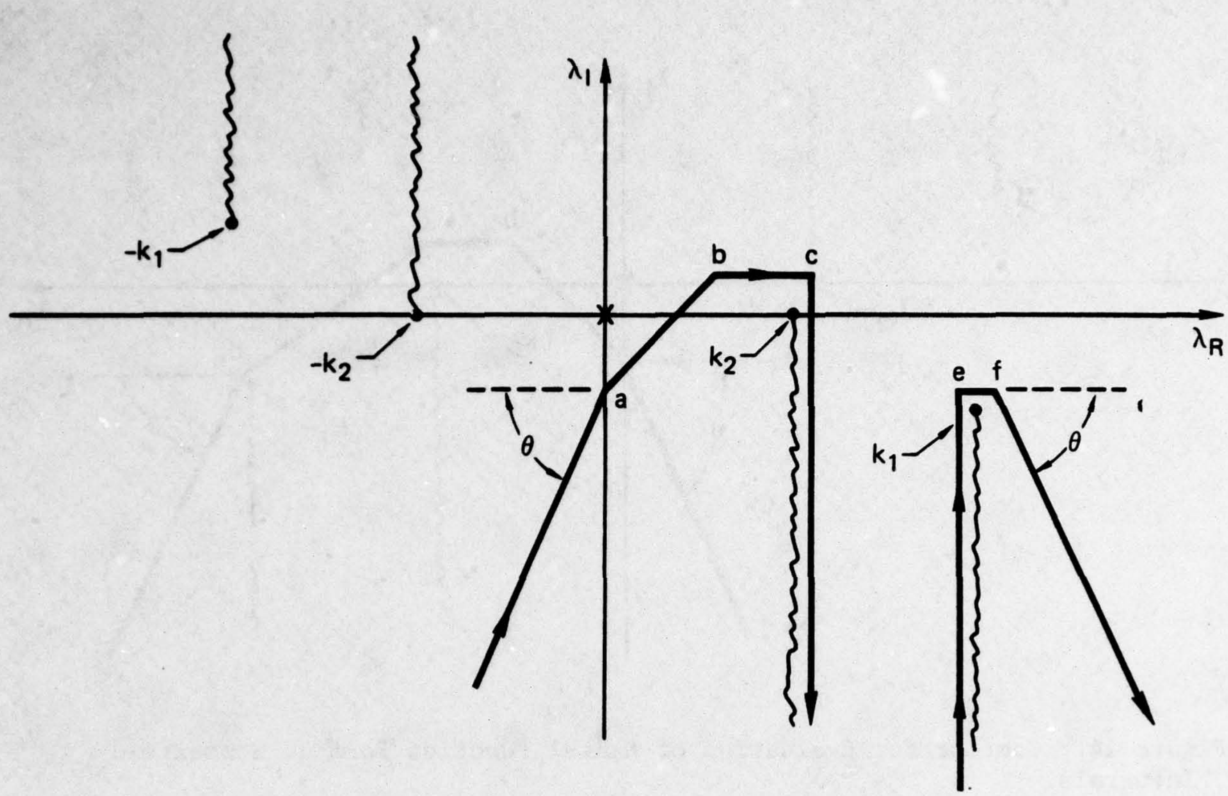


Figure 15. Contour for Hankel Function Form when Real Part k_1 is Large and Imaginary Part k_1 is Small.

integration is applied to successive subsections of length p, where p is the minimum of $0.2\pi/\rho$ or $0.2\pi/(z + z')$, and Shanks' nonlinear transformation (ref. 28) is applied to the sequence of partial sums to accelerate convergence. When ρ and $z + z'$ are both small, the integration interval, p, may be large since the exponential and Bessel functions change slowly and the remaining factors are easily integrated once λ becomes large. For the Bessel function form of the integrals the minimum for $R_1 = [\rho^2 + (z + z')^2]^{1/2}$ is limited only by the maximum number size for the computer. For the Hankel function form the minimum R_1 is about 10^{-5} wavelengths due to the pole at $\lambda\rho = 0$. Either ρ or $z + z'$ may be zero.

The numerical integration results for small R_1 were checked against results from a series approximation (ref. 25) and were in very close agreement. For larger values of R_1 the results from different integration contours were compared as a validation test. Results for the modified Sommerfeld integrals were also checked with normal integrals used in the code WFLLL2A. Earlier studies for the code WFLLL2A, which is capable of computing the field across

the interface, verified the continuity of the computed tangential E field across the interface (ref. 44).

The average time required to evaluate the integrals for a given ρ and $z + z'$ on a CDC 7600 computer is about 0.06 s. Thus about 15 s are required to fill the interpolation grid. Once the grid has been computed and stored, the time to fill an interaction matrix, using interpolation and the Norton formulas, is about four times that for free space.

3. THE IMAGE AND REFLECTION-COEFFICIENT METHODS

The use of a reflected image is a simple and fast way to model the effect of a ground plane. If the ground is perfectly conducting, the structure and its image are exactly equivalent to the structure over the ground. Since use of the image only doubles the time to compute the field, it is always used with a perfect ground. NEC also includes an image approximation for a finitely conducting ground in which the image fields are modified by the Fresnel plane-wave reflection coefficients. Although this is far from exact for a finite ground, it has been shown to provide useful results for structures that are not too near to the ground (refs. 21 and 22). When it can be used, the reflection coefficient method is about twice as fast as the Sommerfeld/Norton method and avoids the need of computing the interpolation grid.

Implementation of the image and reflection coefficient methods in the code is very simple. The Green's function for a perfectly conducting ground is the sum of the free-space Green's function of the source current element and the negative of the free-space Green's function of the image of the source reflected in the ground plane. For the electric field, with free-space Green's dyad $\overline{\overline{G}}(\vec{r},\vec{r}')$ defined in equation (1), the Green's dyad for a perfect ground is

$$\overline{\overline{G}}_{pg}(\vec{r},\vec{r}') = \overline{\overline{G}}(\vec{r},\vec{r}') + \overline{\overline{G}}_I(\vec{r},\vec{r}') , \tag{174}$$

where

$$\overline{\overline{G}}_I(\vec{r},\vec{r}') = -\overline{\overline{I}}_r \cdot \overline{\overline{G}}(\vec{r},\overline{\overline{I}}_r \cdot \vec{r}') , \tag{175}$$

$$\overline{\overline{I}}_r = \hat{x}\hat{x} + \hat{y}\hat{y} - \hat{z}\hat{z} .$$

$\overline{\overline{I}}_r$ is a dyad that produces a reflection in the $z = 0$ plane when used in a dot product. For the magnetic field with free-space Green's dyad

$$\overline{\overline{\Gamma}}(\vec{r},\vec{r}') = \overline{\overline{I}} \times \nabla' g(\vec{r},\vec{r}') \tag{176}$$

the Green's dyad over a perfect ground is

$$\overline{\overline{\Gamma}}pg = \overline{\overline{\Gamma}}(\vec{r},\vec{r}') + \overline{\overline{\Gamma}}_I(\vec{r},\vec{r}') \tag{177}$$

$$\overline{\overline{\Gamma}}_I(\vec{r},\vec{r}') = -\overline{\overline{I}}_r \cdot \overline{\overline{\Gamma}}(\vec{r},\overline{\overline{I}}_r \cdot \vec{r}') \tag{178}$$

The reflection coefficient method for finitely conducting ground uses the image fields modified by the Fresnel reflection coefficients. The Fresnel reflection coefficients, which are strictly correct only for an infinite plane-wave field, depend on the polarization of the incident field with respect to the plane of incidence (i.e., the plane containing the normal to the ground and the vector in the direction of propagation of the wave). The two cases are illustrated in figure 16 where the wave with E in the plane of incidence is termed vertically polarized and E normal to the plane of incidence as horizontally polarized. The Fresnel reflection coefficient for vertically polarized waves is

$$R_V = \frac{\cos\Theta - Z_R\sqrt{1 - Z_R^2\sin^2\Theta}}{\cos\Theta + Z_R\sqrt{1 - Z_R^2\sin^2\Theta}}, \tag{179}$$

where

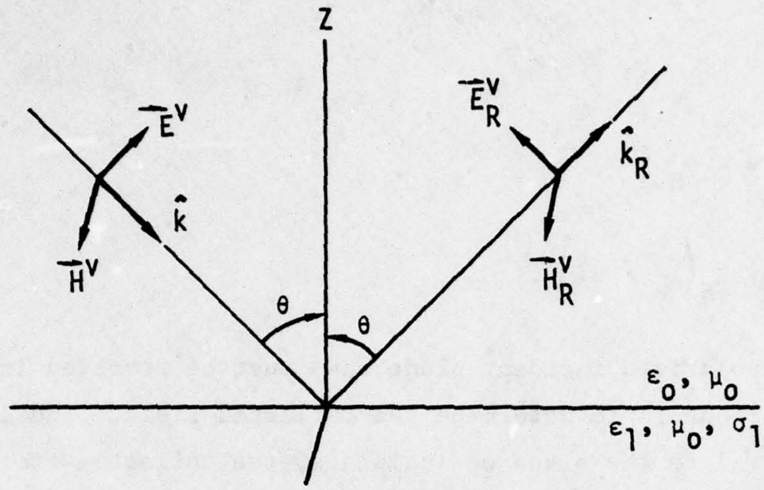
$$\cos\Theta = -\hat{k}\cdot\hat{z},$$

$$Z_R = \left(\frac{\varepsilon_1}{\varepsilon_o} - j\,\frac{\sigma_1}{\omega\varepsilon_o}\right)^{-1/2}.$$

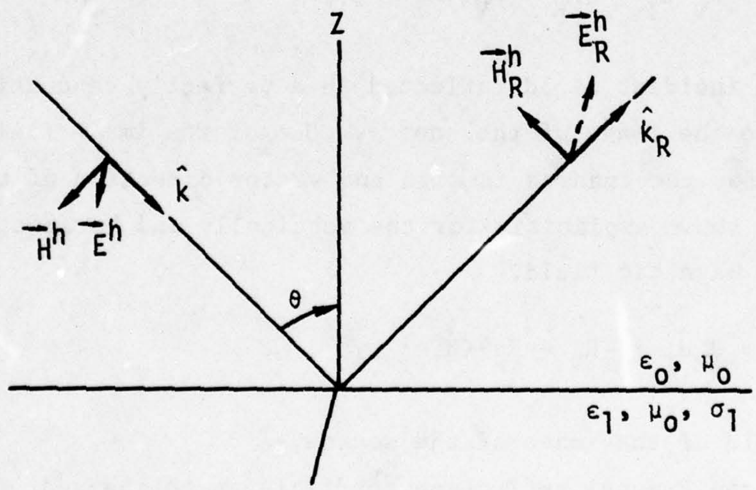
The reflected fields are then

$$\vec{E}_R^V = -R_V\left(\overline{\overline{I}}_R \cdot \vec{E}^V\right),$$

$$\vec{H}_R^V = R_V\,\vec{H}^V.$$



Reflection of Vertically Polarized Wave



Reflection of Horizontally Polarized Wave

Figure 16. Plane-Wave Reflection at an Interface.

For horizontally polarized waves, the reflection coefficient is

$$R_H = \frac{-\left(Z_R \cos \Theta - \sqrt{1 - Z_R^2 \sin^2 \Theta}\right)}{Z_R \cos \Theta + \sqrt{1 - Z_R^2 \sin^2 \Theta}} , \tag{180}$$

and

$$\vec{E}_R^h = - R_H \vec{E}^h ,$$

$$\vec{H}_R^h = R_H \left(\overline{\overline{I}}_R \cdot \vec{H}^h\right) .$$

An arbitrarily polarized incident plane wave must be resolved into horizontal and vertical components to determine the reflected field. Thus, if $\hat{p}$ is the unit vector normal to the plane of incidence, the reflected field due to an incident field $\vec{E}$ is

$$\begin{aligned} \vec{E}_R &= R_H(\vec{E}_I \cdot \hat{p})\hat{p} + R_V \left[\vec{E}_I - (\vec{E}_I \cdot \hat{p})\hat{p}\right] \\ &= R_V \vec{E}_I + (R_H - R_V)(\vec{E}_I \cdot \hat{p})\hat{p} , \end{aligned} \tag{181}$$

where $\vec{E}_I$ is the incident field reflected in a perfectly conducting ground, or the field due to the image of the source. Use of the image field in equation (181) accounts for the changes in sign and vector direction of the incident field that were shown explicitly for the vertically and horizontally polarized cases. For the magnetic field,

$$\vec{H}_R = R_H \vec{H}_I + (R_V - R_H)(\vec{H}_I \cdot \hat{p})\hat{p} , \tag{182}$$

with $\vec{H}_I$ the field of the image of the source.

Applying the Fresnel reflection coefficients to the near fields, the electric field at $\vec{r}$ due to the image of a current element at $\vec{r}\,'$ can be written

$$\begin{aligned} \overline{\overline{G}}_R(\vec{r},\vec{r}\,') &= R_V \overline{\overline{G}}_I(\vec{r},\vec{r}\,') \\ &\quad + (R_H - R_V)\left[\overline{\overline{G}}_I(\vec{r},\vec{r}\,') \cdot \hat{p}\right]\hat{p} , \end{aligned} \tag{183}$$

where

$$\hat{p} = \vec{p}/|\vec{p}| \ ,$$

$$\vec{p} = (\vec{r} - \vec{r}') \times \hat{z} \ ,$$

and $\overline{\overline{G}}_I$ is the Green's function for the image of the source in a perfect ground as defined in equation (175). For magnetic field, the Green's dyad for the modified image is

$$\overline{\overline{\Gamma}}_R(\vec{r},\vec{r}') = R_H \, \overline{\overline{\Gamma}}_I(\vec{r},\vec{r}') + (R_V - R_H)\left[\overline{\overline{\Gamma}}_I(\vec{r},\vec{r}') \cdot \hat{p}\right]\hat{p} \ . \tag{184}$$

The Green's functions for electric and magnetic fields over an imperfectly conducting ground, resulting from the reflection coefficient approximation are then

$$\overline{\overline{G}}_g(\vec{r},\vec{r}') = \overline{\overline{G}}(\vec{r},\vec{r}') + \overline{\overline{G}}_R(\vec{r},\vec{r}') \ , \tag{185}$$

$$\overline{\overline{\Gamma}}_g(\vec{r},\vec{r}') = \overline{\overline{\Gamma}}(\vec{r},\vec{r}') + \overline{\overline{\Gamma}}_R(\vec{r},\vec{r}') \ . \tag{186}$$

Use of the Green's function's $\overline{\overline{G}}_g$ and $\overline{\overline{\Gamma}}_g$ results in a straightforward extension of the EFIE and MFIE for structures over an imperfect ground.

NEC also includes a reflection coefficient approximation for a radial wire ground screen, as used by Miller and Deadrick (ref. 29). This is based on an approximation developed by Wait (ref. 30) for the surface impedance of the radial-wire ground screen on an imperfectly conducting ground, as the parallel combination of the surface impedance, ζ_1, of the ground plane

$$\zeta_1 = \left(\frac{j\mu_o\omega}{\sigma_1 + j\epsilon_1\omega}\right)^{1/2} \ ,$$

and an approximate surface impedance Z_g of the ground screen

$$Z_g(\rho) = \frac{j\mu_o\omega\rho}{N} \, \ell n\left(\frac{\rho}{NC_o}\right) \ .$$

The ground screen impedance assumes a parallel wire grid having the wire spacing that the radial wires have at a distance ρ from the center. N is the number of radial wires in the screen, and C_o is the radius of the wires. The surface impedance of the ground screen on an imperfect ground is then

$$\zeta_e = \frac{\zeta_1 Z_g}{\zeta_1 + Z_g} .$$

From the definition of surface impedance,

$$E_{tangential} = \zeta_e H_{tangential}$$

at the surface. Using the fact that E and H in the incident wave are related by η the free-space impedance, reflection coefficients are derived as

$$R_H = \frac{\eta - \zeta_e \cos \Theta}{\eta + \zeta_e \cos \Theta} ,$$

and

$$R_V = \frac{\eta \cos \Theta - \zeta_e}{\eta \cos \Theta + \zeta_e} .$$

This is the form the Fresnel reflection coefficients take when the index of refraction is large compared to unity ($|Z_R|^2 << 1$). This condition is satisfied in most realistic problems; furthermore, the surface-impedance boundary condition is a valid approximation only when the refractive index of the ground is large compared to unity. The surface impedance is used in conjunction with the reflection coefficient method previously discussed to provide an approximate model of a radial-wire ground screen.

Due to the assumption of specular reflection, only the properties of the ground directly under a vertical antenna will affect its current distribution. At the origin of the radial-wire ground screen, the impedance is zero (Z_g is not allowed to be negative) so the impedance and current distribution of a vertical antenna at the origin will be the same as over a perfect conductor. The far fields, however, will demonstrate the effect of the screen as the specular point moves away from the origin. For antennas other than the vertical

antenna, it should be pointed out that the inherent polarization sensitivity of the screen (i.e., E parallel or perpendicular to the ground wires) has not been considered in this approximation. When limited accuracy can be accepted this ground screen approximation provides a large time saving over explicit modeling with the Sommerfeld/Norton method since the ground screen does not increase the number of unknowns in the matrix equation.

Section V
Modeling of Antennas

Previous sections have dealt with the problem of determining the current induced on a structure by an arbitrary excitation. We now consider some specific problems in modeling antennas and scatterers, including models for a voltage source on a wire, lumped and distributed loads, nonradiating networks, and transmission lines. Calculations of some observable quantities are also covered including input impedance, radiated field, and antenna gain.

1. SOURCE MODELING

The approach used in NEC is applicable to a number of electromagnetic analysis problems. For receiving antennas and EMP studies, the excitation is the field of an incident plane wave and the desired response is the induced current at one or more points on the structure. In scattering analysis the excitation is still an incident plane wave, but the desired response is the field radiated by the induced currents. In the case of a wire transmitting antenna, however, the excitation is generally a voltage source on the wire. The antenna source problem has received a considerable amount of attention in the literature. A rather thorough exposition on the appropriate source configuration for the linear dipole antenna has been given by King (ref. 31). The delta-function source, which may be visualized as an infinitesimally thin, circumferential belt of axially directed electric field [or, alternatively, as a frill of magnetic current at the antenna feed point (ref. 32)], is convenient mathematically, but of somewhat questionable physical realizability. Since the excitation can be specified only at discrete points in NEC, a delta-function source is not feasible.

A useful source model, however, is an electric field specified at a single match point. For a voltage source of strength V on segment i, the element in the excitation vector corresponding to the applied electric field at the center of segment i is set to

$$E_i = \frac{V}{\Delta_i} , \qquad (187)$$

where Δ_i is the length of segment i. The direction of E_i is toward the positive end of the voltage source so that it pushes charge in the same direction as the source. The field at other match points is set to zero.

The actual effective voltage is the line integral of the applied field along the wire. This cannot be determined beforehand since the field is known only at segment centers, but can be determined after the solution for current by integrating the scattered field produced by the current. For equal length segments in the vicinity of the source this field, which must be the negative of the applied field at every point on the wire, is nearly constant over segment i and drops sharply at the segment ends. This results in an actual voltage of approximately $\Delta_i E_i$ as assumed in equation (187). When the source segment and adjacent segments are not of equal length, however, the actual voltage, obtained by integrating the scattered field, may differ from the intended value.

Ideally, this source model applies a voltage V between the ends of the source segment. Hence, the antenna input admittance could be computed as the current at the segment ends or, in an unsymmetric case, the average of the current at the two ends, divided by the applied voltage. In practice the segment is sufficiently short so that the current variation over its length is small and the current at the center can be used rather than the ends. When segment lengths in the source region are unequal, the computed input admittance may be inaccurate due to the discrepancy between the actual and assumed voltages. Use of the actual voltage, obtained by integrating the near field, will generally give an accurate admittance although it will require additional effort for computation.

An alternate source model that is less sensitive to the equality of segment lengths in the source region is based on a discontinuity in the derivative of current. This source model is similar to one used by Andreasen and Harris (ref. 33), and its use in a program similar to NEC was reported by Adams, Poggio, and Miller (ref. 24). For this model, the source region is viewed as a biconical transmission line with feed point at the source location, as illustrated in figure 17. The voltage between a point at s and the symmetric point on the other side of the line is then related to the derivative of the current by the transmission line equation,

$$V(s) = -\, jZ_o \frac{\partial I(ks)}{\partial (ks)} , \tag{188}$$

where Z_o is the characteristic impedance of the transmission line. The characteristic impedance of a biconical transmission line of half-angle Θ is

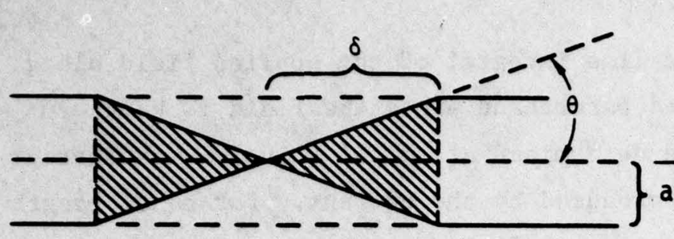


Figure 17. Biconical Transmission Line Model of Source Region.

$$Z_o = 120 \ \ln \left(\cot \frac{\Theta}{2} \right) ,$$

or for small angles,

$$Z_o \approx 120 \ \ln \left(\frac{2}{\Theta} \right) . \tag{189}$$

For a source on a wire, however, the proper choice for δ in figure 17, defining the angle Θ, is unclear. Adams at al. (ref. 24) used an average value of Z_o obtained by averaging equation (189) for δ ranging from zero to d as

$$Z_{avg} = \frac{1}{d} \int_o^d 120 \ \ln \left(\frac{2\delta}{a} \right) d\delta$$

$$= 120 \left[\ln \left(\frac{2d}{a} \right) - 1 \right] ,$$

where d is set equal to the distance from the source location at the segment end to the match point at the segment center. The voltage across the line is then

$$V(s) = - j \ 120 \left[\ln \left(\frac{2d}{a} \right) - 1 \right] \frac{\partial I(ks)}{\partial (ks)} .$$

Allowing for a current unsymmetric about the source, the voltage V_o of a source at s_o is related to a discontinuity in current derivative as

$$\lim_{\varepsilon \to o} \left[\left. \frac{\partial I(ks)}{\partial (ks)} \right|_{s = s_o + \varepsilon} - \left. \frac{\partial I(ks)}{\partial (ks)} \right|_{s = s_o - \varepsilon} \right] =$$

$$\frac{-jV_o}{60 \left[\ln \left(\frac{2d}{a} \right) - 1 \right]} . \tag{190}$$

This discontinuity in current derivative is introduced into NEC by modifying the current expansion on the wire. The normal expansion for N_s wire segments is

$$I(s) = \sum_{j=1}^{N_s} \alpha_j \, f_j(s) \ ,$$

where the basis functions, f_j, are defined in section III-1 such that I(s) has continuous value and derivative along wires, and satisfies Kirchoff's law and a condition on charge density at junctions.

For a current-slope-discontinuity source at the first end of segment ℓ, the current expansion is modified to

$$I(s) = \sum_{j=1}^{N_s} \alpha_j \, f_j(s) + \beta_\ell \, f_\ell^*(s) \ , \tag{191}$$

where f_ℓ^* is a basis function for segment ℓ, as defined in section III-1, but computed as if the first end of segment ℓ were a free end and the segment radius were zero. Hence, f_ℓ^* goes to zero with nonzero derivative at the source location.

If f_ℓ^* on segment ℓ is

$$f_\ell^*(s) = A_\ell^* + B_\ell^* \sin k(s - s_\ell) + C_\ell^* \cos k(s - s_\ell)$$

$$\times \, |s - s_\ell| < \Delta_\ell/2 \ ,$$

then

$$\frac{\partial}{\partial(ks)} f_\ell^*(s) \Big|_{s = s_\ell - \Delta_\ell/2} = B_\ell^* \cos(k\Delta_\ell/2) + C_\ell^* \sin(k\Delta_\ell/2) \ .$$

Since the sum of the normal basis functions has continuous value and derivative at $s = s_\ell - \Delta_\ell/2$, the current in equation (191) has a discontinuity in derivative of

$$\lim_{\varepsilon \to 0} \left\{ \frac{\partial}{\partial(ks)} I(s) \Big|_{s=s_\ell - \Delta_\ell/2 + \varepsilon} - \frac{\partial}{\partial(ks)} I(s) \Big|_{s=s_\ell - \Delta_\ell/2 - \varepsilon} \right\} =$$

$$\beta_\ell \left\{ B_\ell^* \cos(k\Delta_\ell/2) + C_\ell^* \sin(k\Delta_\ell/2) \right\} .$$

Hence, from equation (190), a source voltage of V_o requires a value of β_ℓ in the current expansion of

$$\beta_\ell = \frac{-jV_o}{60}\left\{\left[\ell n\left(\frac{\Delta_\ell}{a_\ell}\right) - 1\right]\left[B_\ell^* \cos(k\Delta_\ell/2) + C_\ell^* \sin(k\Delta_\ell/2)\right]\right\}^{-1} . \tag{192}$$

The linear system for the current expansion constants, obtained by substituting equation (191) for f in equation (18), is

$$\sum_{j=1}^{N_s} \alpha_j <w_i, Lf_j> = <w_i, e> - \beta_\ell <w_i, f_\ell^*> . \tag{193}$$

$$i = 1, \ldots N_s$$

In matrix notation, corresponding to equation (19),

$$[G]\ [A] = [E] + \beta_\ell\ [F] , \tag{194}$$

where F_i is the excitation for segment or patch equation number i due to the field of f_ℓ^*, and E_i is the excitation for segment or patch equation number i from other sources (if there are any). The interaction matrix G is independent of this source as it is of other sources. The solution for the expansion coefficients is then

$$[A] = [G^{-1}]\left\{[E] + \beta_\ell\ [F]\right\} ,$$

where A supplies the coefficients α_j in equation (191) to determine the current. This method is easily extended to several sources. The modified basis function f_ℓ^* appears to introduce an asymmetry into the current, but this is not the case since the other basis function amplitudes are free to adjust accordingly.

The current-slope-discontinuity source results in an effective applied field that is much more localized in the source region than that of the constant-field source defined by equation (187). The difference is shown in near-field plots for the two source models in figure 18, taken from Adams

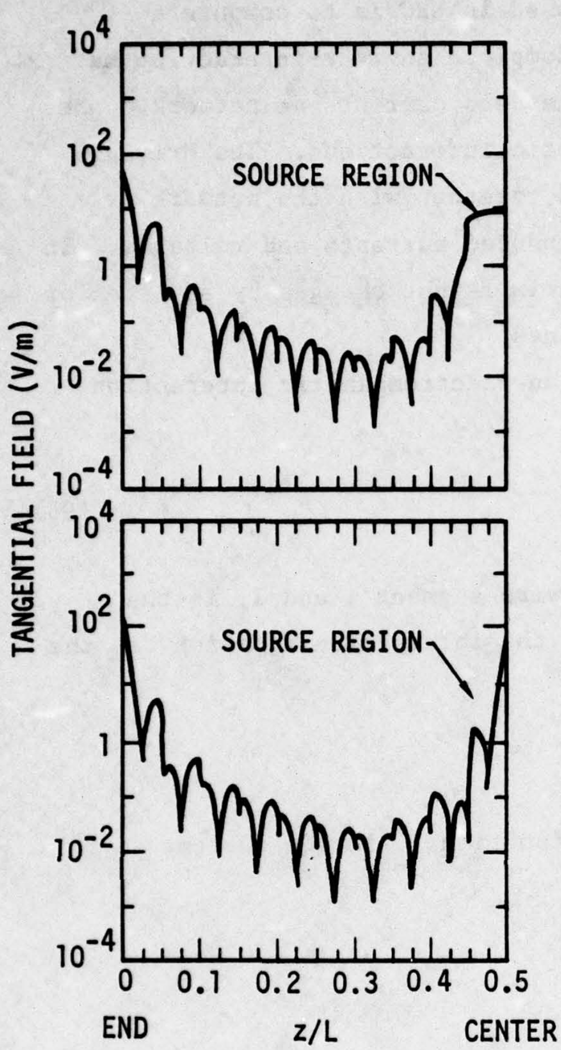


Figure 18. Field Plots for a Linear Dipole, Ω=15.

et al. (ref. 24). The near fields are for a half wavelength dipole antenna with $\Omega = 15$ [$\Omega = 2\ell n(L/a)$, L = length, a = radius] and with 10 segments on half of the antenna covered by the plots. The constant-field source is seen to result in a nearly rectangular field distribution in the source region while the field of the slope-discontinuity source approaches a delta function. The integrals of these two source-field distributions yield approximately the same voltages, however.

With the slope-discontinuity-source model, the input admittance is the ratio of the current at the segment end, where the source is located, to the source voltage. Adams et al. also present results showing the effect on admittance of varying the source-segment length relative to the lengths of adjacent segments, showing that the slope-discontinuity source is much less sensitive to segment length than is the constant-field source. The two segments on opposite sides of the source must have equal lengths and radii, however. For very short segment lengths, the slope-discontinuity model may break down although, as with the constant-field source, the correct admittance can be obtained by integrating the near field to obtain the source voltage.

2. NONRADIATING NETWORKS

Antennas often include transmission lines, lumped circuit networks, or a combination of both connecting between different parts or elements. When the currents on transmission lines or at network ports are balanced, the resulting fields cancel and can often be neglected, greatly simplifying the

modeling problem. The solution procedure used in NEC is to compute a driving-point-interaction matrix from the complete segment-interaction matrix. The driving-point matrix relates the voltages and currents at network connection points as required by the electromagnetic interactions. The driving-point-interaction equations are then solved together with the network or transmission line equations to obtain the induced currents and voltages. In this way the larger segment-interaction matrix is not changed by addition or modification of networks or transmission lines.

The solution described below assumes an electromagnetic interaction matrix equation of the form,

$$[G]\,[I] = -\,[E]\ , \tag{195}$$

where E_i is the exciting electric field on wire segment i and I_i is the current at the center of segment i. In NEC the interaction equation has the form,

$$[G]\,[A] = -\,[E]\ ,$$

where A_i is the amplitude of the i^{th} basis function f_i in the current expansion,

$$I(s) = \sum_{i=1}^{N_s} A_i\, f_i(s)\ .$$

The same solution technique can be used, however, by computing I from A whenever I is needed. This must be done in computing the elements of the inverse of G, G^{-1}_{ij}, which below represent the current on segment i due to a unit field on segment j.

A model consisting of N_s segments will be assumed with a general M-port network connected to segments 1 through M. The network is described by the admittance equations,

$$\sum_{j=1}^{M} Y_{ij}\, V_j = I^t_i \qquad i=1,\ \ldots,\ M\ , \tag{196}$$

where V_i and I^t_i are the voltage and current at port i, with reference directions as shown in figure 19.

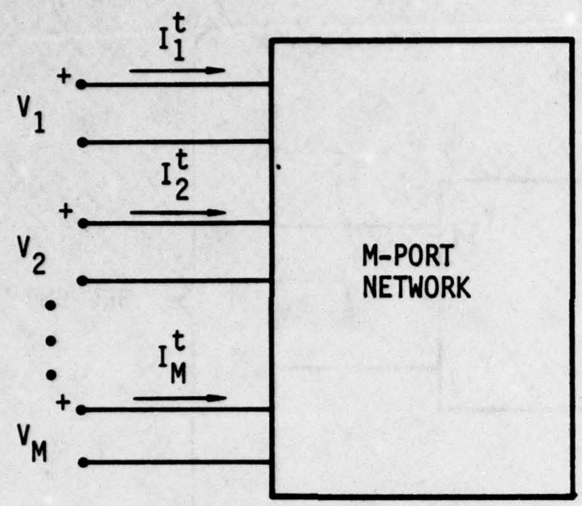


Figure 19. Voltage and current Reference Directions at Network Ports.

The connection of a network port to a segment is illustrated in figure 20. The segment is broken, and the port is connected so that

$$I_i^t = - I_i \ , \tag{197}$$

where I_i is the segment current. Figure 21 shows a voltage source of strength V_i connected across the network port at segment i. In this case,

$$I_i^t = I_i^g - I_i \ . \tag{198}$$

In either case, the port voltage may be related to the applied field on the segment by the constant-field voltage source model of equation (187).

We will assume that segments 1 through M_1 are connected to network ports without voltage sources, and segments $M_1 + 1$ through M are connected to network ports with voltage sources. The remaining segments have no network

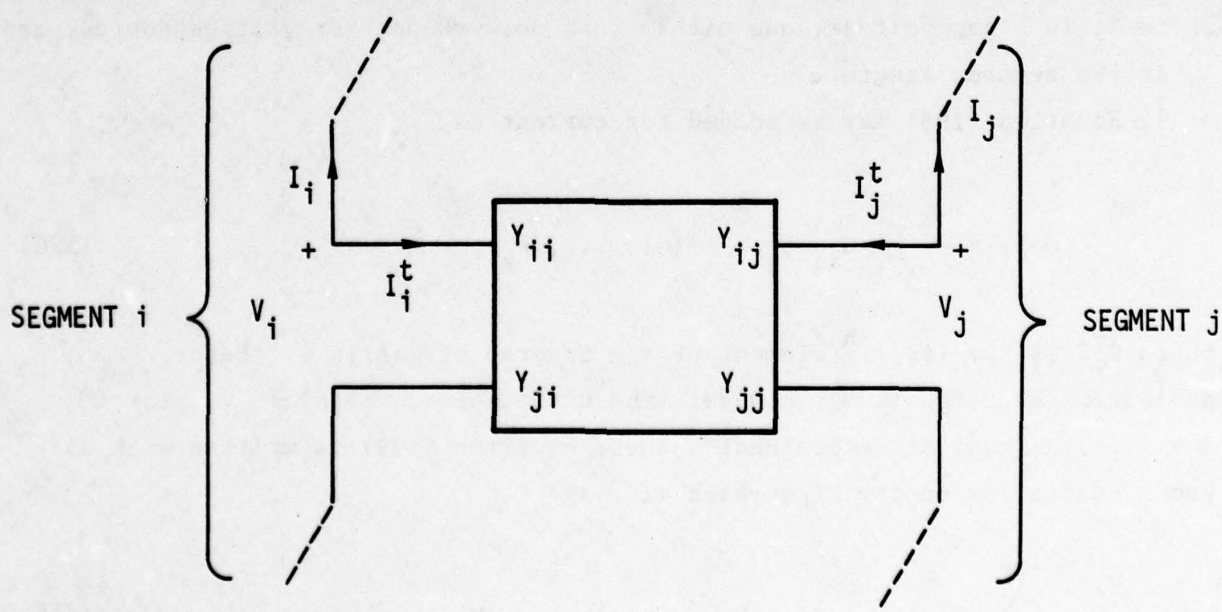


Figure 20. Network Connection to Segments.

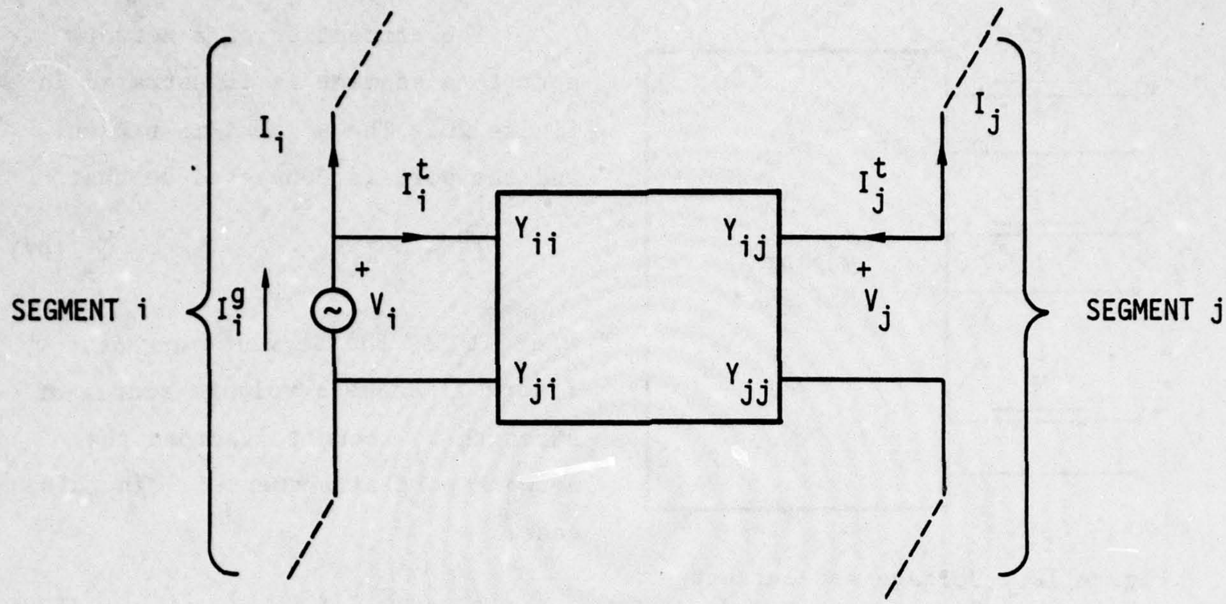


Figure 21. Network Port and Voltage Source Connected to a Segment.

connections but may have voltage sources. In addition all of the segments may be excited by an incident field represented by E_i^I on segment i. The total field on segment i is then

$$E_i = \frac{V_i}{\Delta_i} + E_i^I ,$$

where V_i is a gap voltage, due either to a network port or voltage source, and Δ_i is the segment length.

Equation (195) may be solved for current as

$$I_i = - \sum_{j=1}^{N_s} G_{ij}^{-1} E_j \qquad i=1, \ldots, N_s , \tag{199}$$

where G_{ij}^{-1} is the $(i,j)^{th}$ element of the inverse of matrix G. Before evaluating equation (199), however, the unknown port voltages, V_i, for $i = 1,\ldots,M_1$ must be determined. Hence, equation (199) is written with all known quantities on the right-hand side as

$$\sum_{j=1}^{M_1} G_{ij}^{-1} E_j^P + I_i = B_i \qquad i=1, \ldots, M_1 , \tag{200}$$

where

$$E_j^P = \frac{V_j}{\Delta_j} ,$$

and

$$B_i = - \sum_{j=1}^{M_1} G_{ij}^{-1} E_j^I - \sum_{j=M_1+1}^{N_s} G_{ij}^{-1} E_j .$$

Similarly, the network equations (196) are written using equation (197) as

$$\sum_{j=1}^{M_1} Y'_{ij} E_j^P + I_i = C_i \qquad i=1, \ldots, M_1 , \tag{201}$$

where

$$Y'_{ij} = \Delta_j Y_{ij} ,$$

$$C_i = - \sum_{j=M_1+1}^{M} Y_{ij} V_j .$$

The current is then eliminated between equations (200) and (201) to yield

$$\sum_{j=1}^{M_1} \left(G_{ij}^{-1} - Y'_{ij} \right) E_j^P = B_i - C_i \qquad i=1, \ldots, M_1 . \tag{202}$$

The solution procedure is then to solve equation (202) for E_j^P for $j = 1,\ldots,M_1$. Then, with the complete excitation vector determined, use equation (199) to determine I_i for $i = 1,\ldots,N_s$. Finally, the remaining network equations with equation (198) are used to compute the generator currents as

$$I_i^g = \sum_{j=1}^{M} Y_{ij} V_j + I_i \qquad i=M_1+1, \ldots M . \tag{203}$$

The currents I_i^g determine the input admittances seen by the sources.

In NEC the general M-port network used here is restricted to multiple two-port networks, each connecting a pair of segments.

3. TRANSMISSION LINE MODELING

Transmission lines interconnecting parts of an antenna may be modeled either explicitly by including the transmission line wires in the thin-wire model, or implicitly by the method described in the preceding section for nonradiating networks. For an implicit model, the short-circuit-admittance parameters of the transmission line viewed as a two-port network are

$$Y_{11} = Y_{22} = - j \, Y_o \cot(k\ell) \; ,$$

$$Y_{12} = Y_{21} = j \, Y_o \csc(k\ell) \; ,$$

where Y_o is the characteristic admittance of the line, k is the wave number $(2\pi/\lambda)$, and ℓ is the length of the line. If a separate admittance element is connected across the end of a transmission line, its admittance is added to the self-admittance of that network port.

The implicit model is limited, however, in that it neglects interaction between the transmission line and the antenna and its environment. This approximation is justified if the currents in the line are balanced, i.e., in a log periodic dipole antenna, and in general if the transmission line lies in an electric symmetry plane. The balance can be upset, however, if the transmission line is connected to an unbalanced load or by unsymmetric interactions. If the unbalance is significant, the transmission line can be modeled explicitly by including the wires in the thin-wire model. The explicit model is completely general, and yields accurate results since the sine, cosine, and constant current expansion in NEC is a good representation of the sinusoidal transmission line currents. The accuracy is demonstrated in figure 22 for transmission lines terminated in short circuit, open circuit, and matched loads.

The explicit transmission line model is, of course, less efficient in computer time and storage because of the additional segments required. In cases where the physical line presence does have a significant effect on the results, the effect may be modeled by explicitly modeling a single conductor of the line while using the implicit model to represent the balanced current component.

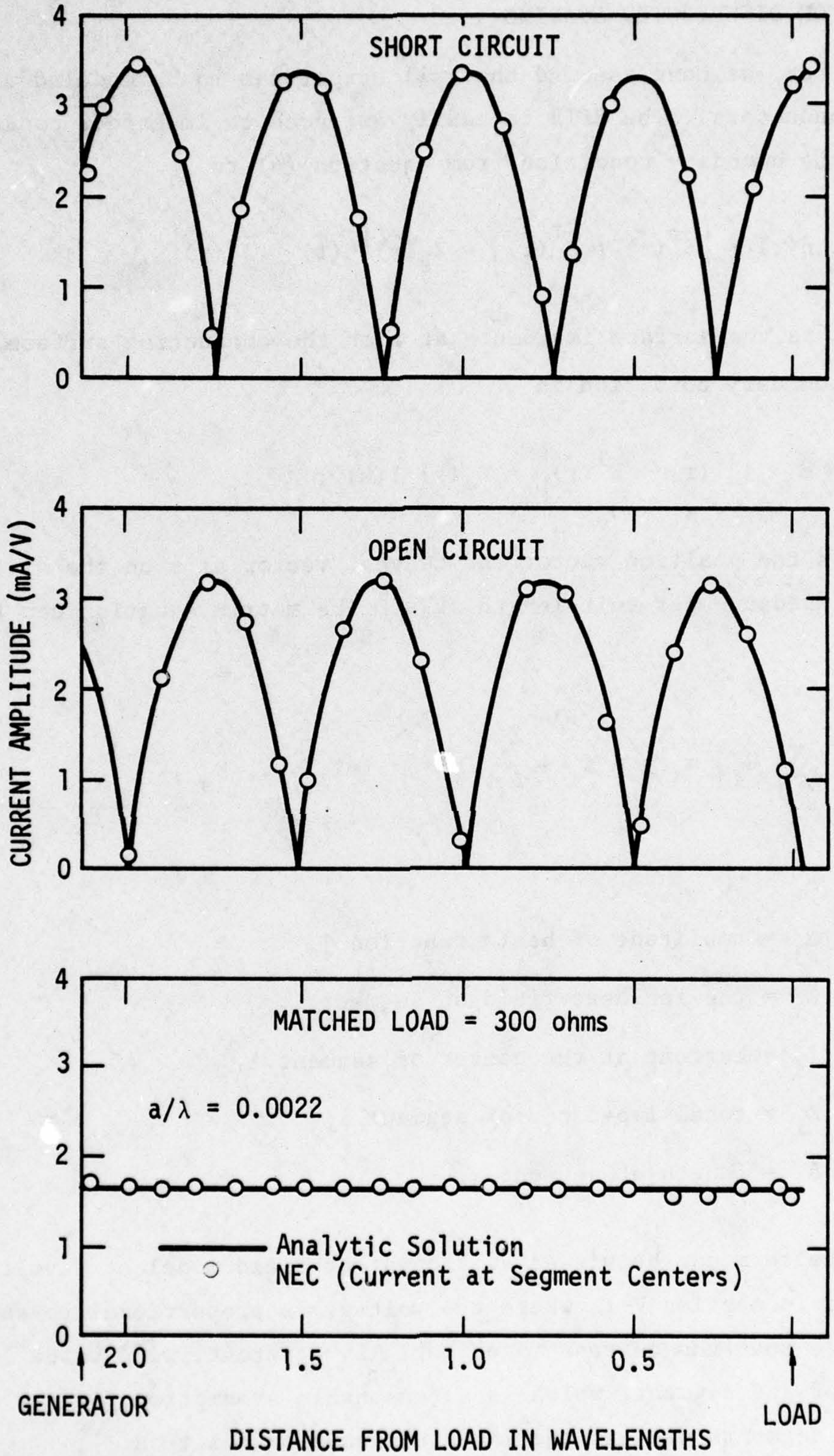


Figure 22. Current Distribution on a Two-Wire Transmission Line from NEC Compared with the Ideal Transmission Line Solution.

4. LUMPED OR DISTRIBUTED LOADING

Thus far, we have assumed that all structures to be modeled are perfect electric conductors. The EFIE is easily extended to imperfect conductors by modifying the boundary condition from equation (4) to

$$\hat{n}(\vec{r}) \times \left[\vec{E}^{s}(\vec{r}) + \vec{E}^{I}(\vec{r})\right] = Z_s(\vec{r}) \left[\hat{n}(\vec{r}) \times \vec{J}_s(\vec{r})\right] ,$$

where $Z_s(\vec{r})$ is the surface impedance at $\vec{r}$ on the conducting surface. For a wire, the boundary condition is

$$\hat{s} \cdot \left[\vec{E}^{s}(\vec{r}) + \vec{E}^{I}(\vec{r})\right] = Z_w(s)\, I(s) ,$$

with $\vec{r}$ and $\hat{s}$ the position vector and tangent vector at s on the wire and $Z_w(s)$ the impedance per-unit-length at s. The matrix equation can then be written,

$$\sum_{j=1}^{N_s} G_{ij}\, \alpha_j = -\, E_i + \frac{Z_i}{\Delta_i} I_i \qquad i=1, \ldots, N_s , \tag{204}$$

where

α_j = amplitude of basis function j,

E_i = the incident field on segment i,

I_i = current at the center of segment i,

Z_i = total impedance of segment i,

Δ_i = length of segment i.

The impedance term can be viewed as a constant field model of a voltage source, as described in section V-1, where the voltage is proportional to current. It is assumed that the current is essentially constant, with value I_i, over the length of the segment, which is a reasonable assumption for the electrically short segments used in the integral equation solution.

The impedance term can be combined with the matrix by expressing I_i in terms of the α_j as

$$I_i = \sum_{j=1}^{N_s} \alpha_j \left(A_j^i + C_j^i\right),$$

where A_j^i and C_j^i are the coefficients of the constant and cosine terms, respectively, in the section of basis function j extending onto segment i. If basis function j does not extend onto segment i, then A_j^i and C_j^i are zero. The matrix equation modified by loading is then

$$\sum_{j=1}^{N_s} G'_{ij} \alpha_j = - E_i \qquad i=1, \ldots, N_s , \tag{205}$$

where

$$G'_{ij} = G_{ij} - \frac{Z_i}{\Delta_i} \left(A_j^i + C_j^i\right). \tag{206}$$

For a lumped circuit element, Z_i is computed from the circuit equations. For a distributed impedance, Z_i represents the impedance of a length Δ_i of wire, which in the case of a round wire of finite conductivity is given by

$$Z_i = \frac{j\Delta_i}{a_i} \sqrt{\frac{\omega\mu}{2\pi\sigma}} \left[\frac{\mathrm{Ber}(q) + j\, \mathrm{Bei}(q)}{\mathrm{Ber}'(q) + j\, \mathrm{Bei}'(q)}\right],$$

where

$q = (\omega\mu\sigma)^{1/2}\, a_i$,

a_i = wire radius,

σ = wire conductivity,

Ber, Bei = Kelvin functions.

This expression takes account of the limited penetration of the field into an imperfect conductor.

5. RADIATED FIELD CALCULATION

The radiated field of an antenna or reradiated field of a scatterer can be computed from the induced current by using a simplified form of equation (1) valid far from the current distribution. The far-field approximation, valid when the distance from the current distribution to the observation

point is large compared to both the wavelength and the dimensions of the current distribution, treats the distance $|\vec{r} - \vec{r}'|$ as constant within the integral except in the phase term, $\exp(-jk|\vec{r} - \vec{r}'|)$. For a structure consisting of a wire portion with contour L and current distribution $\vec{I}(s)$, and a surface portion S with current $\vec{J}_s(\vec{r})$, the far-zone field is

$$\vec{E}(\vec{r}_o) = \frac{jk\eta}{4\pi} \frac{\exp(-jkr_o)}{r_o}$$

$$\times \left\{ \int_L \left[\left(\hat{k} \cdot \vec{I}(s) \right) \hat{k} - \vec{I}(s) \right] \exp(j\vec{k}\cdot\vec{r}) ds \right. \tag{207}$$

$$\left. + \int_S \left[\left(\hat{k}\cdot\vec{J}_s(\vec{r}) \right) \hat{k} - \vec{J}_s(\vec{r}) \right] \exp(j\vec{k}\cdot\vec{r}) dA \right\} ,$$

where $\vec{r}_o$ is the position of the observation point $\hat{k} = \vec{r}_o/|\vec{r}_o|$, $k = 2\pi/\lambda$, and $\vec{k} = k\hat{k}$. The first integral can be evaluated in closed form over each straight wire segment for the constant, sine, and cosine components of the basis functions, and reduces to a summation over the wire segments. With the surface current on each patch represented by a delta function at the patch center, the second integral becomes a summation over the patches.

The radiation pattern of an antenna can be computed by exciting the antenna with a voltage source and using equation (207) to compute the radiated field for a set of directions in space. Alternatively, since the transmitting and receiving patterns are required by reciprocity to be the same, the pattern can be determined by exciting the antenna with plane-waves incident from the same directions and computing the currents at the source point. The solution procedure in NEC does not guarantee reciprocity, however, since the different expansion and weighting functions may produce asymmetry in the matrix. Large differences between the receiving and transmitting patterns or a significant lack of reciprocity in bistatic scattering are indications of inaccuracy in the solution, possibly from too coarse a segmentation of the wires or surfaces.

The power gain of an antenna in the direction specified by the spherical coordinates (Θ,ϕ) is defined as

$$G_p(\Theta,\phi) = 4\pi \frac{P(\Theta,\phi)}{P_{in}} ,$$

where $P(\Theta,\phi)$ is the power radiated per unit solid angle in the direction (Θ,ϕ), and P_{in} is the total power accepted by the antenna from the source. P_{in} is computed from the voltage and current at the source as

$$P_{in} = \frac{1}{2}\,\mathrm{Re}(VI^*) ,$$

and

$$P(\Theta,\phi) = \frac{1}{2} R^2\, \mathrm{Re}(\vec{E}\times\vec{H}^*) = \frac{R^2}{2\eta}\,(\vec{E}\cdot\vec{E}^*) .$$

$\vec{E}$ is obtained from equation (207) with $\vec{r}_o$ in the direction (Θ,ϕ), and $r_o = R$. Directive gain is similarly defined as

$$G_d(\Theta,\phi) = 4\pi\,\frac{P(\Theta,\phi)}{P_{rad}} ,$$

where P_{rad} is the total power radiated by the antenna,

$$P_{rad} = P_{in} - P_{loss} ,$$

and P_{loss} is the total ohmic loss in the antenna.

The radiated field of an antenna over ground is modified by the ground interaction, as discussed in section IV. When the range from the antenna to the observer, R, approaches infinity, the Sommerfeld formulation for the field reduces exactly to a direct field determined by equation (207) and a field from the image modified by the Fresnel reflection coefficient. In some cases, however, when the observer is at a finite distance from the antenna, the field components proportional to $1/R^2$ may be significant. While the 1/R terms are generally much larger than the $1/R^2$ terms at practical observation distances from an antenna, the 1/R terms vanish at grazing angles over an imperfect ground plane leaving only the $1/R^2$ terms, dominated by a term known as the ground wave. The ground wave is, of course, included in Sommerfeld's expressions. Norton's asymptotic approximations (ref. 26) are used, however, since they are more easily evaluated and give adequate accuracy. Norton's formulas, which are in Part II of this manual under subroutine GWAVE, are valid for R greater than a few wavelengths and to second order in k_1^2/k_2^2. When the ground wave is included, the field has radial as well as transverse components.

6. ANTENNA COUPLING

Coupling between antennas is often a parameter of interest, especially when a receiving system must be protected from a nearby transmitter. Maximum power transfer between antennas occurs when the source impedance and receiver load impedance are conjugate-matched to their antennas. Determination of this condition is complicated by the antenna interaction, however, since the input impedance of one antenna depends on the load connected to the other antenna. NEC-2 includes an algorithm for determining the matched loads and maximum coupling by a method that was added to special versions of the previous codes NEC-1 and AMP (ref. 34).

The coupling problem can be solved in closed form by the Linville method (ref. 35), a technique used in rf amplifier design. The first step is to determine the two-port admittance parameters for the coupled antennas by exciting each antenna with the other short-circuited and computing the self and mutual admittances from the currents computed by NEC. The maximum coupling is then

$$G_{MAX} = \frac{1}{L}\left[1 - (1-L^2)^{1/2}\right],$$

where

$$L = \frac{|Y_{12}Y_{21}|}{2\mathrm{Re}(Y_{11})\,\mathrm{Re}(Y_{22}) - \mathrm{Re}(Y_{12}Y_{21})}.$$

The matched load admittance on antenna 2 for maximum coupling is

$$Y_L = \left[\frac{1-\rho}{1+\rho} + 1\right]\mathrm{Re}(Y_{22}) - Y_{22},$$

where

$$\rho = \frac{G_{MAX}(Y_{12}Y_{21})^*}{|Y_{12}Y_{21}|},$$

and the corresponding input admittance of antenna 1 is

$$Y_{IN} = Y_{11} - \frac{Y_{21}Y_{12}}{Y_L + Y_{22}}.$$

References

1. Miller, E. K. et al., *Final Technical Report on the Log-Periodic Scattering Array Program*, MBAssociates Report No. MB-R-70/105, Contract No. F04701-68-C-0188, 1970.
2. *RCS Computer Program BRACT, Log-Periodic Scattering Array Program - II*, MBAssociates Report No. MB-R-69/46, Contract No. F04701-68-C-0188, 1969.
3. *Antenna Modeling Program - Composite User's Manual*, MBAssociates Report No. MB-R-74/46, 1974.
4. *Antenna Modeling Program - Engineering Manual*, MBAssociates Report No. MB-R-74/62, 1974.
5. *Antenna Modeling Program - Systems Manual*, MBAssociates Report No. MB-R-74/62, 1974.
6. *Antenna Modeling Program - Supplementary Computer Program Manual (AMP2)*, MBAssociates Report No. MB-R-75/4, 1975.
7. *Numerical Electromagnetic Code (NEC-1) Part III: NEC User's Guide*, Lawrence Livermore Laboratory, 1977.
8. *Numerical Electromagnetic Code (NEC-1) Part II: NEC Program Description - Code*, Lawrence Livermore Laboratory, 1977.
9. Albertsen, N. C., Hansen, J. E., and Jensen, N. E., *Computation of Spacecraft Antenna Radiation Patterns*, The Technical University of Denmark, Lyngby, Denmark, June 1972.
10. Poggio, A. J. and Miller, E. K., "Integral Equation Solutions of Three-Dimensional Scattering Problems," Chapt. IV in *Computer Techniques for Electromagnetics*, edited by R. Mittra, Pergamon Press, New York, 1973.
11. Poggio, A. J. and Adams, R. W., *Approximations for Terms Related to the Kernel in Thin-Wire Integral Equations*, UCRL-51985, Lawrence Livermore Laboratory, CA, December 19, 1975.
12. Van Bladel, J., *Electromagnetic Fields*, McGraw-Hill, New York, 1964.
13. Harrington, R. F., *Field Computation by Moment Methods*, MacMillan, New York, 1968.
14. Yeh, Y. S. and Mei, K. K., "Theory of Conical Equiangular Spiral Antennas," *Part I - Numerical Techniques, IEEE Trans. Ant. and Prop., AP-15*, 1967, p. 634.

15. Neureuther, A. R. et al., "A Comparison of Numerical Methods for Thin Wire Antennas," *Presented at the 1968 Fall URSI Meeting,* Dept. of Electrical Engineering and Computer Sciences, University of California, Berkeley, 1968.

16. Miller, E. K. et al., *An Evaluation of Computer Programs Using Integral Equations for the Electromagnetic Analysis of Thin Wire Structures,* UCRL-75566, Lawrence Livermore Laboratory, CA, March 1974.

17. Wu, T. T. and King, R. W. P., "The Tapered Antenna and Its Application to the Junction Problem for Thin Wires," *IEEE Trans. Ant. and Prop., AP-24,* No. 1, pp. 42-45, January 1976.

18. Poggio, A. J., *Integral Representations for Fields Due to Sources on Open Surfaces with Applications to End Caps,* UCRL-51723, Lawrence Livermore Laboratory, CA, December 16, 1974.

19. Ralston, A., *A First Course in Numerical Analysis,* McGraw-Hill, New York 1965.

20. Sommerfeld, A., *Partial Differential Equations in Physics,* Academic Press, New York, 1964.

21. Burke, G. J., Miller, E. K., and Poggio, A. J., *Dielectric Ground Interactions with a Log-Periodic Antenna,* Final Report, ECOM-0285-F, U.S. Army Electronics Command, Fort Monmouth, N.J., 1971 (contract DAAB07-70-C-0285).

22. Miller, E. K., Poggio, A. J., Burke, G. J., and Selden, E. S., *Analysis of Wire Antennas in the Presence of a Conducting Half Space: Part I. The Vertical Antenna in Free Space. Part II. The Horizontal Antenna in Free Space,* UCRL-73217, Lawrence Livermore Laboratory, CA, 1972. (Also in *Canadian J. Phys.*)

23. Lager, D. L. and Lytle, R. J., *Fortran Subroutines for the Numerical Evaluation of Sommerfeld Integrals Unter Anderem,* UCRL-51821, Lawrence Livermore Laboratory, CA, May 21, 1975.

24. Adams, R. W., Poggio, A. J., and Miller, E. K., *Study of a New Antenna Source Model,* UCRL-51693, Lawrence Livermore Laboratory, CA, October 28, 1974.

25. Brittingham, J. N., Miller, E. K., and Okada, J. T., *SOMINT: An Improved Model for Studying Conducting Objects Near Lossy Half-Spaces,* UCRL-52423, Lawrence Livermore Laboratory, CA, February 24, 1978.

26. Norton, K. A., "The Propagation of Radio Waves Over the Surface of the Earth and in the Upper Atmosphere," *Proceedings of the Institute of Radio Engineers,* Vol. 26, No. 9, Sept. 1937.

27. Lytle, R. J. and Lager, D. L., *Numerical Evaluation of Sommerfeld Integrals,* UCRL-51688, Lawrence Livermore Laboratory, CA, October 23, 1974.

28. Shanks, D., "Non-Linear Transformations of Divergent and Slowly Convergent Sequences," *J. Math. Phys.,* 24, 1 (1955).

29. Miller, E. K. and Deadrick, F. J., *Computer Evaluation of Loran-C Antennas,* UCRL-51464, Lawrence Livermore Laboratory, CA, Oct. 17, 1973.

30. Wait, J. R., "Characteristics of Antennas Over Lossy Earth," *Antenna Theory, Part II,* R. E. Collin and F. J. Zucker, Eds., McGraw-Hill, New York, pp. 386-437, 1969.

31. King, R. W., *The Theory of Linear Antennas,* Harvard Univ. Press, Cambridge, Massachusetts, 1956.

32. Otto, D. V., "The Admittance of Cylindrical Antennas Driven from a Coaxial Line," *Radio Sci.,* Vol. 2, No. 9, pp. 1031-1042, 1967.

33. Andreasen, M. G. and Harris, F. G., *Analyses of Wire Antennas of Arbitrary Configuration by Precise Theoretical Numerical Techniques,* Tech. Rept. ECOM 0631-F, Granger Associates, Palo Alto, California, Contract DAAB07-67-C-0631, 1968.

34. Logan, J. C. and Rockway, J. W., *Computer Techniques for Shipboard Topside Antenna Modeling Below UHF,* TN-3284, Naval Electronics Laboratory Center, November 17, 1976.

35. Rubin, D., *The Linville Method of High Frequency Transistor Amplifier Design,* Naval Weapons Center, Research Department, NWCCL TP 845, Corona Laboratories, Corona, California, March 1969.

RAC/gw

Part II: Program Description - Code

Preface

The Numerical Electromagnetics Code (NEC) has been developed at the Lawrence Livermore Laboratory, Livermore, California, under the sponsorship of the Naval Ocean Systems Center and the Air Force Weapons Laboratory. It is an advanced version of the Antenna Modeling Program (AMP) developed in the early 1970's by MBAssociates for the Naval Research Laboratory, Naval Ship Engineering Center, U.S. Army ECOM/Communications Systems, U.S. Army Strategic Communications Command, and Rome Air Development Center under Office of Naval Research Contract N00014-71-C-0187. The present version of NEC is the result of efforts by G. J. Burke and A. J. Poggio of Lawrence Livermore Laboratory.

The documentaton for NEC consists of three parts:

Part I: NEC Program Description - Theory

Part II: NEC Program Description - Code

Part III: NEC User's Guide

The documentation has been prepared by using the AMP documents as foundations and by modifying those as needed. In some cases this led to minor changes in the original documents while in many cases major modifications were required.

Over the years many individuals have been contributors to AMP and NEC and are acknowledged here as follows:

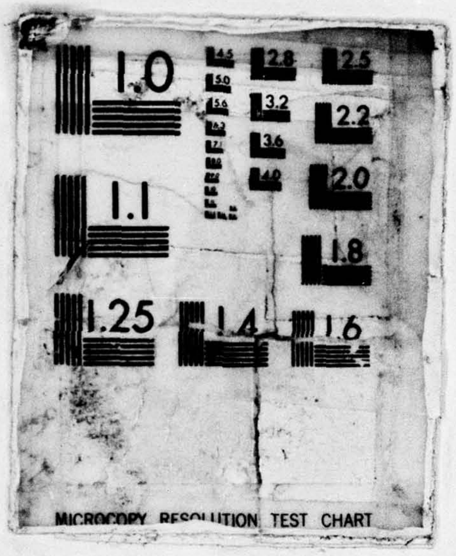
R. W. Adams	R. J. Lytle
J. N. Brittingham	E. K. Miller
G. J. Burke	J. B. Morton
F. J. Deadrick	G. M. Pjerrou
K. K. Hazard	A. J. Poggio
D. L. Knepp	E. S. Selden
D. L. Lager	

The support for the development of NEC-2 at the Lawrence Livermore Laboratory has been provided by the Naval Ocean Systems Center under MIPR-N0095376MP. Cognizant individuals under whom this project was carried out include: J. Rockway and J. Logan. Previous development of NEC also included the support of the Air Force Weapons Laboratory (Project Order 76-090) and was monitored by J. Castillo and TSgt. H. Goodwin.

Work was performed under the auspices of the U.S. Department of Energy under contract No. W-7405-Eng-48. Reference to a company or product name

AD-A075 289 NUMERICAL ELECTROMAGNETIC CODE (NEC)-METHOD OF MOMENTS A USER-ORIENTED CO.. (U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA G J BURKE ET AL. 18 JUL 77 NOSC/TD-116-VOL-1 2/6

UNCLASSIFIED AFWL-TR-76-320-VOL-1 F/G 12/5 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

Contents

Section		Page

Section | Page

List of Illustrations

Abstract

The Numerical Electromagnetics Code (NEC-2) is a computer code for analyzing the electromagnetic response of an arbitrary structure consisting of wires and surfaces in free space or over a ground plane. The analysis is accomplished by the numerical solution of integral equations for induced currents. The excitation may be an incident plane wave or a voltage source on a wire, while the output may include current and charge density, electric or magnetic field in the vicinity of the structure, and radiated fields. Hence, the code may be used for antenna analysis or scattering and EMP studies.

This document is Part II of a three-part report. It contains a detailed description of the Fortran coding, including the definitions of variables and constants, and a listing of the code. The other two documents cover the equations and numerical methods (Part I) and instructions for use of the code (Part III).

KEY WORDS FOR DD FORM 1473:

EM scattering

EMP

Wire Model

Method of moments

Section I
Introduction

The Numerical Electromagnetics Code (NEC-2)* is a user-oriented computer code for the analysis of the electromagnetic response of antennas and other metal structures. It is built around the numerical solution of integral equations for the currents induced on the structure by sources or incident fields. This approach avoids many of the simplifying assumptions required by other solution methods and provides a highly accurate and versatile tool for electromagnetic analysis.

The code combines an integral equation for smooth surfaces with one specialized to wires to provide for convenient and accurate modeling of a wide range of structures. A model may include nonradiating networks and transmission lines connecting parts of the structure, perfect or imperfect conductors, and lumped-element loading. A structure may also be modeled over a ground plane that may be either a perfect or imperfect conductor.

The excitation may be either voltage sources on the structure or an incident plane wave of linear or elliptic polarization. The output may include induced currents and charges, near electric or magnetic fields, and radiated fields. Hence, the program is suited to either antenna analysis or scattering and EMP studies.

This document is Vol. II of a three-part report on NEC. It contains a detailed description of the Fortran coding. Section II contains for each routine: (1) a statement of purpose, (2) a narrative description of the methodology, (3) definitions of variables and constants, and (4) a listing of the code. The remaining sections cover the common blocks, system library functions, array dimension limitations, and subroutine linkage.
The information in Vol. II will be of use mainly to persons attempting to modify the code or to use it on a computer system with which the delivered deck is not compatible.

Vol. I describes the equations and numerical methods used in NEC and Vol. III contains instructions for using the code, including preparation of input data and interpretation of output. Persons attempting to use NEC for the first time should start by reading Vol. III. Vol. I will help the new user to understand the capabilities and limitations of NEC.

*NEC-2 will be abbreviated to NEC elsewhere in this volume.

Section II
Code Description

In this section, each routine in NEC is described in detail. The main program is described first and is followed by the subroutines in alphabetical order. For each routine, there is a brief statement of its purpose, a description of the code, an alphabetized listing and definition of important variables and constants, and a listing of the code. Variables that are in common blocks, and hence occur in several routines, are usually omitted from the lists for individual routines. They are defined in Section III under their common block labels.

Following line MA 495 in the main program, all quantities of length have been normalized to wavelength. Current is normalized to wavelength throughout the solution. This changes the appearance of many of the equations. In particular the wave number, $k = 2\pi/\lambda$, usually appears as 2π.

MAIN

PURPOSE

To handle input and output and to call the appropriate subroutines.

METHOD

The structure of MAIN is shown in the flow charts of Figures 1 and 2 where Figure 1 represents the first half of the code to about line MA 459. Comment cards are read and printed after line MA 72 and subroutine DATAGN is called at MA 90 to read and process structure data. If a Numerical Green's Function (NGF) file was read in DATAGN then subroutine FBNFG is called to determine whether file storage is needed for the matrix and to allocate core storage. When a NGF has not been read the mode of matrix storage cannot be determined until line MA 464 since it depends on whether a NFG file is to be written.

The box labeled "Read data card" in Figure 1 refers to the READ statement at MA 139. Any of the types of data cards in Table 1 may be read at this point to set parameters or to request execution of the solution part of the code.

The integer variables IGØ and IFLOW are keys to the operation of the code. IGØ indicates the stage of completion of the solution as listed in Table 2. When a card requesting execution is read (NE, NH, RP, WG, or XQ) the solution part of the code (Figure 2) is entered at the point determined by IGØ (see MA 385, MA 420, MA 429, and MA 457). After the current has been computed IGØ is given the value five. If subsequent data cards change parameters, the value of IGØ is reduced to the value in Table 1 to indicate the point beyond which the solution must be repeated. For example, when an EX card is read IGØ is set equal to three if it was greater than three but is not changed if it was less than three. For cards that request execution "ex." is shown in Table 1.

IFLOW is used to indicate the type of the previous data card. When several cards of the same type can be used together (CP, LD, NT, TL, and EX for voltage sources) a counter is incremented and data is added to arrays if the card is the same as the previous card as indicated by IFLOW. If the previous card was different the counter is initialized and previous data in the arrays is destroyed. IFLOW is also used to indicate what type of card

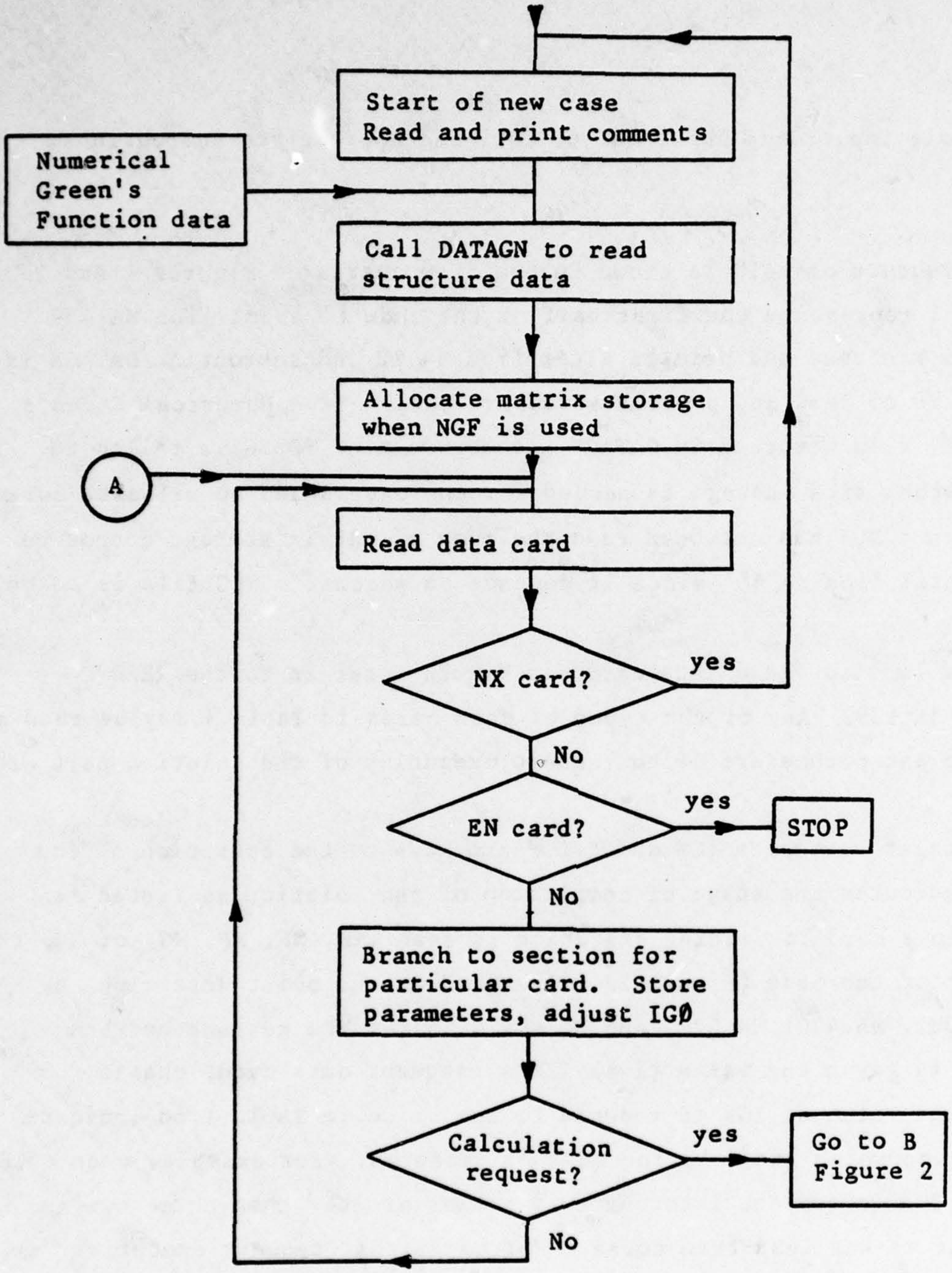


Figure 1. Flow Diagram of Main Program Input Section

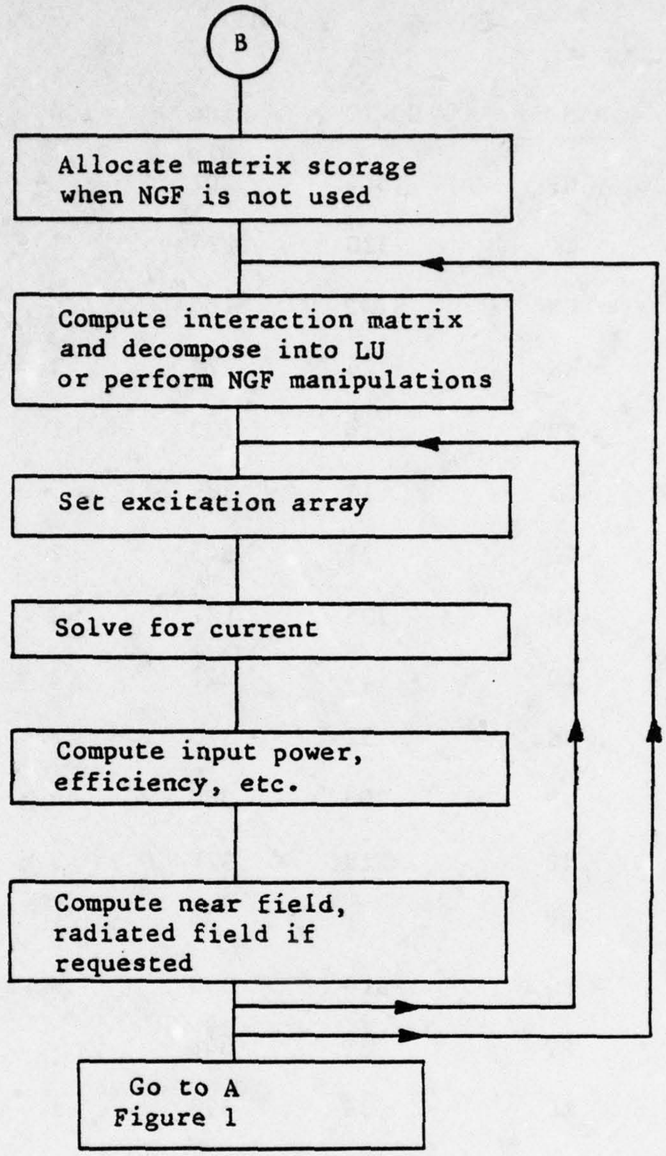


Figure 2. Flow Diagram of Main Program
Computation Section

TABLE 1

	I	AIN(I)	GO TO	Line	IGØ	IFLOW
1	21	CP	304	202	-	2
2	19	EK	320	194	2	1
3	13	EN	STØP	166	-	-
4	5	EX	24	275	3	5
5	2	FR	16	172	1	1
6	9	GD	34	389	-	9
7	4	GN	21	245	2	4
8	16	KH	305	187	2	1
9	3	LD	17	221	2	3
10	8	NE	32	370	ex.*	8
11	17	NH	208	368	ex.*	8
12	6	NT	28	321	3	6
13	12	NX	1	69	1	1
14	18	PQ	319	358	-	-
15	15	PT	31	348	-	-
16	10	RP	36	398	ex.	10
17	14	TL	28	321	3	6
18	20	WG	322	424	ex.	12
19	7	XQ	37	433	ex.	7 or 11

* NE and NH do not cause execution when multiple frequencies have been requested on the FR card. This allows computation of both near fields and radiated fields in a frequency loop.

Table 2.

IGO	Completion Point
1	Start
2	Frequency has been set and geometry scaled to wavelength
3	Interaction matrix filled and factored
4,5	Current computed and printed

requested the solution (NE, RP, etc.). Cards such as RP may be stacked together but are not stored since they are acted upon as they are encountered.

The solution part of the code contains a loop over frequency starting at MA 463 and a loop over incident field direction starting at MA 562. FBLOCK is called at MA 465 to determine whether file storage is required for the matrix. From MA 466 to MA 493 the structure data are scaled from units of meters to wavelength or from one wavelength to the next when frequency is changed. Subroutine LOAD is called at MA 497 to fill array ZARRAY for the given frequency. At MA 520 the Sommerfeld interpolation tables are read from file TAPE21 if this option is used. NXA(1) is set to zero at MA 67 so the test ensures that the tape is read only once.

When the NGF option is not in use the matrix is filled by subroutine CMSET at MA 537 and factored by subroutine FACTRS at MA 540. When the NGF is used the equivalent steps are performed by CMNGF and FACGF. If a NGF file is to be written, subroutine GFOUT is called at MA 557 to write TAPE20.

Subroutine ETMNS, called at MA 582, fills the excitation array and the current is computed in subroutine NETWK called at MA 611. If transmission lines or two port networks are used NETWK combines the network equations with driving-point interaction equations derived from the primary interaction matrix. Otherwise the current is computed directly from the primary matrix.

The remainder of MAIN prints the currents and calls subroutines for near fields, radiated fields or coupling.

SYMBOL DICTIONARY:

AIN	= mnemonic from data card
ATST	= array of possible data card mnemonics
CMAG	= magnitude of the current in amperes
COM	= array to store text from comment cards
CURI	= current on segment I in amperes
CVEL	= (velocity of light) (10^{-6}) in meters/second
DELFRQ	= frequency increment (additive or multiplicative)
DPH	= far-field ϕ angle increment in degrees (input quantity)
DTH	= far-field θ angle increment in degrees (input quantity)

DXNR, DYNR, DZNR	= near-field observation point increments (input quantities with multiple meanings -- see NE card)
EPH	= current component in direction $\hat{t}_2$ on patch
EPHA	= phase angle of EPH
EPHM	= magnitude of EPH
EPSC	= complex dielectric constant of ground $\varepsilon_c = \varepsilon_r - j\sigma/\omega\varepsilon_0$.)
EPSCF	= ε_c read from file TAPE21
EPSR	= ε_r
EPSR2	= ε_r for outer ground region
ETH	= current component in direction $\hat{t}_1$ on patch
ETHA	= phase angle of ETH
ETHM	= magnitude of ETH
EX	= $\hat{x}$ component of current on a patch
EXTIM	= time at start of run (seconds)
EY	= $\hat{y}$ component of current on a patch
EZ	= $\hat{z}$ component of current on a patch
FJ	= $\sqrt{-1}$
FMHZ	= frequency in MHz
FMHZS	= frequency in MHz
FNORM	= multiply used array; stores impedances for printing of the normalized impedance or stores currents in the receiving pattern case for printing normalized receiving pattern
FR	= (next frequency)/(present frequency)
FR2	= (FR)(FR)
GNOR	= if non-zero, equals gain normalization factor (dB) from RP card
HPOL	= array containing polarization types (Hollerith)
IAVP	= input integer flag used in average gain logic (RP card)
IAX	= input integer flag specifying gain type (RP card)
IB11	= location in array CM for start of storage of submatrix B when NGF is used
IC11	= location in array CM for start of storage of submatrix C when NGF is used

ID11	=	location in CM for submatrix D
IEXK	=	flag to select the extended thin-wire kernel
IFAR	=	input integer flag specifying type of field calculation and type of ground system in far field (RP card)
IFLOW	=	integer flag used to distinguish various input sections
IFRQ	=	input integer flag specifying type of frequency stepping (FR card)
IGO	=	integer to indicate stage of completion of the solution
INC	=	incident field loop index
INOR	=	input integer flag used for normalized gain request (RP card)
IPD	=	input integer flag selects gain type for normalization (RP card)
IPED	=	input integer flag used for impedance normalization request (EX card)
IPTAG	=	input integer for print control equal to segment tag number (PT card)
IPTAGF	=	input integer for print control specifying segment placement in a set of equal tags (PT card)
IPTAGT	=	same function as IPTAGF (input, PT card)
IPTFLG	=	input integer flag specifying type of print control (PT card)
IPTAQ, IPTAQF, IPTAQT, IPTFLQ	=	same as above four variables but for PQ card
IRESRV	=	length of array CM in complex numbers
IRNGF	=	storage in array CM that is reserved for later use when a NGF file is written
ISANT	=	array of segment numbers for voltage sources
ISAVE	=	segment number for normalized receiving pattern calculation

ISEG1 (I), ISEG2 (I)	=	segment numbers of end 1 and end 2 of the i^{th} network connection
ITMP1 to ITMP5	=	temporary storage
IX	=	array for matrix pivot element information
IX11	=	location in CM of the start of an array in the NGF solution
IXTYP	=	excitation type from EX card
KCOM	=	number of comment cards read
LDTAG	=	tag number of loaded segment
LDTAGF	=	number of first loaded segment in set of segments having given tag
LDTAGT	=	last loaded segment
LDTYP	=	loading type
LOADMX	=	maximum number of loading cards
MASYM	=	flag to request matrix asymmetry calculation
MHZ	=	frequency loop index
MPCNT	=	counter for data cards
NCOUP	=	number of excitation points for coupling calculation
NCSEG, NCTAG	=	excitation segment for coupling calculation
NEAR	=	increment option for near field points
NEQ	=	order of the primary interaction matrix
NEQ2	=	number of new unknowns in NGF mode
NETMX	=	maximum number of network data cards
NFEH	=	0 for near E field, 1 for near H
NFRQ	=	number of frequency steps
NONET	=	number of network data cards
NORMF	=	dimension of FNORM
NPHI	=	number of phi steps in incident field
NPHIC	=	loop index for phi in incident field
NPRINT	=	print control flag for subroutine NETWK
NRX, NRY, NRZ	=	number of steps in near field evaluation loops
NSANT	=	number of voltage sources
NSMAX	=	maximum number of voltage sources

MAIN

NTHI	=	number of theta steps in incident field
NTHIC	=	loop index for theta in incident field
PH	=	phase angle of current or charge (degrees)
PHISS	=	initial ϕ value for incident field
PIN	=	P_{in} = total power supplied to a structure by all voltage sources (Σ Re(VI*)/2). For a Hertzian dipole source $P_{in} = \eta(\pi/3)\lvert I\ell/\lambda\rvert^2$.
PLOSS	=	power lost in distributed and point structure loads in watts
PNET	=	array contains Hollerith transmission line type
RFLD	=	if non-zero, equal to input far-field observation distance in meters
RKH	=	minimum separation for use of approximate interaction equations
SCRWLT	=	input length of radials in radial wire screen (GN card) in meters
SCRWRT	=	radius of wires in radial wire ground screen in meters
SIG	=	conductivity of ground (σ in mhos/meter on GN card)
SIG2	=	conductivity of second medium in mhos/meter (GN and GD card)
TA	=	$\pi/180$
THETIS	=	initial θ for incident field
THETS	=	initial θ for radiated field
TIM	=	matrix computation time (seconds)
TMP1 to TMP6	=	temporary input variables
XPR1 to XPR6	=	input quantities for incident field or Hertzian dipole illumination
ZLC, ZLI, ZLR	=	input quantities for loading
ZPNORM	=	impedance normalization quantity

CONSTANTS

1.E-20	=	used as small value test

1.745329252	=	$\pi/180$
2367.067	=	$2\pi\eta_0$
59.96	=	$1/(2\pi c\varepsilon_0)$
299.8	=	$c/10^6$

```
      PROGRAM NEC(INPUT,TAPE5=INPUT,OUTPUT,TAPE11,TAPE12,TAPE13,TAPE14,
     1TAPE15,TAPE16,TAPE20,TAPE21)
C
C     NUMERICAL ELECTROMAGNETICS CODE (NEC2)  DEVELOPED AT LAWRENCE
C     LIVERMORE LAB., LIVERMORE, CA.  (CONTACT G. BURKE, 415-422-8414)
C     FILE CREATED 4/11/80.
C
C                 ***********NOTICE**********
C     THIS COMPUTER CODE MATERIAL WAS PREPARED AS AN ACCOUNT OF WORK
C     SPONSORED BY THE UNITED STATES GOVERNMENT.  NEITHER THE UNITED
C     STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY, NOR ANY OF
C     THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR
C     THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR
C     ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY,
C     COMPLETENESS OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT
C     OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT
C     INFRINGE PRIVATELY-OWNED RIGHTS.
C
      INTEGER  AIN,ATST,PNET,HPOL
      COMPLEX  CM,FJ,VSANT,ETH,EPH,ZRATI,CUR,CURI,ZARRAY,ZRATI2
      COMPLEX  EX,EY,EZ,ZPED,VQD,VQDS,T1,Y11A,Y12A,EPSC,U,U2,XX1,XX2
      COMPLEX  AR1,AR2,AR3,EPSCF,FRATI
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),
     1SI(300),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),
     2ITAG(300),ICONX(300),WLAM,IPSYM
      COMMON /CMB/CM(4000)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,
     1ICASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      COMMON/SAVE/IP(600),KCOM,COM(13,5),EPSR,SIG,SCRWLT,SCRWRT,FMHZ
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),
     1 CII(300),CUR(900)
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF
      COMMON/YPARM/NCOUP,ICOUP,NCTAG(5),NCSEG(5),Y11A(5),Y12A(20)
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,
     1IPCON(10),NPCON
      COMMON/VSORC/VQD(30),VSANT(30),VQDS(30),IVQD(30),ISANT(30),
     1IQDS(30),NVQD,NSANT,NQDS
      COMMON/NETCX/ZPED,PIN,PNLS,NEQ,NPEQ,NEQ2,NONET,NTSOL,NPRINT,
     1MASYM,ISEG1(30),ISEG2(30),X11R(30),X11I(30),X12R(30),X12I(30),
     1X22R(30),X22I(30),NTYP(30)
      COMMON/FPAT/NTH,NPH,IPD,IAVP,INOR,IAX,THETS,PHIS,DTH,DPH,
     1RFLD,GNOR,CLT,CHT,EPSR2,SIG2,IXTYP,XPR6,PINR,PNLR,PLOSS,
     1NEAR,NFEH,NRX,NRY,NRZ,XNR,YNR,ZNR,DXNR,DYNR,DZNR
      COMMON /GGRID/ AR1(11,10,4),AR2(17,5,4),AR3(9,8,4),EPSCF,DXA(3),
     1DYA(3),XSA(3),YSA(3),NXA(3),NYA(3)
      COMMON/GWAV/U,U2,XX1,XX2,R1,R2,ZMH,ZPH
      DIMENSION CAB(1),SAB(1),X2(1),Y2(1),Z2(1)
      DIMENSION LDTYP(30),LDTAG(30),LDTAGF(30),LDTAGT(30),ZLR(30),
     1ZLI(30),ZLC(30)
      DIMENSION ATST(21),PNET(6),HPOL(3),IX(600)
      DIMENSION FNORM(200)
      DIMENSION T1X(1),T1Y(1),T1Z(1),T2X(1),T2Y(1),T2Z(1)
      EQUIVALENCE (CAB,ALP),(SAB,BET),(X2,SI),(Y2,ALP),(Z2,BET)
      EQUIVALENCE (T1X,SI),(T1Y,ALP),(T1Z,BET),(T2X,ICON1),(T2Y,ICON2),
     1 (T2Z,ITAG)
      DATA ATST/2HCE,2HFR,2HLD,2HGN,2HEX,2HNT,2HXQ,2HNE,2HGD,2HRP,2HCM,
     1 2HNX,2HEN,2HTL,2HPT,2HKH,2HNH,2HPQ,2HEK,2HWG,2HCP/
      DATA HPOL/6HLINEAR,5HRIGHT,4HLEFT/
      DATA PNET/6H      ,2H  ,6HSTRAIG,2HHT,6HCROSSE,1HD/
      DATA TA/1.745329252E-02/,CVEL/299.8/
      DATA LOADMX,NSMAX,NETMX/30,30,30/,NORMF/200/
```

```
      CALL SECOND(EXTIM)                                                MA  64
      FJ=(0.,1.)                                                        MA  65
      LD=300                                                            MA  66
      NXA(1)=0                                                          MA  67
      IRESRV=4000                                                       MA  68
1     KCOM=0                                                            MA  69
2     KCOM=KCOM+1                                                       MA  70
      IF (KCOM.GT.5) KCOM=5                                             MA  71
      READ(5,125)AIN,(COM(I,KCOM),I=1,13)                               MA  72
      IF(KCOM.GT.1)GO TO 3                                              MA  73
      PRINT 126                                                         MA  74
      PRINT 127                                                         MA  75
      PRINT 128                                                         MA  76
3     PRINT 129, (COM(I,KCOM),I=1,13)                                   MA  77
      IF (AIN.EQ.ATST(11)) GO TO 2                                      MA  78
      IF (AIN.EQ.ATST(1)) GO TO 4                                       MA  79
      PRINT 130                                                         MA  80
      STOP                                                              MA  81
4     CONTINUE                                                          MA  82
      DO 5 I=1,LD                                                       MA  83
5     ZARRAY(I)=(0.,0.)                                                 MA  84
      MPCNT=0                                                           MA  85
      IMAT=0                                                            MA  86
C                                                                       MA  87
C     SET UP GEOMETRY DATA IN SUBROUTINE DATAGN                         MA  88
C                                                                       MA  89
      CALL  DATAGN                                                      MA  90
      IFLOW=1                                                           MA  91
      IF(IMAT.EQ.0)GO TO 326                                            MA  92
C                                                                       MA  93
C     CORE ALLOCATION FOR ARRAYS B, C, AND D FOR N.G.F. SOLUTION        MA  94
C                                                                       MA  95
      NEQ=N1+2*M1                                                       MA  96
      NEQ2=N-N1+2*(M-M1)+NSCON+2*NPCON                                  MA  97
      CALL FBNGF(NEQ,NEQ2,IRESRV,IB11,IC11,ID11,IX11)                   MA  98
      GO TO 6                                                           MA  99
326   NEQ=N+2*M                                                         MA 100
      NEQ2=0                                                            MA 101
      IB11=1                                                            MA 102
      IC11=1                                                            MA 103
      ID11=1                                                            MA 104
      IX11=1                                                            MA 105
      ICASX=0                                                           MA 106
6     NPEQ=NP+2*MP                                                      MA 107
      PRINT 135                                                         MA 108
C                                                                       MA 109
C     DEFAULT VALUES FOR INPUT PARAMETERS AND FLAGS                     MA 110
C                                                                       MA 111
      IGO=1                                                             MA 112
      FMHZS=CVEL                                                        MA 113
      NFRQ=1                                                            MA 114
      RKH=1.                                                            MA 115
      IEXK=0                                                            MA 116
      IXTYP=0                                                           MA 117
      NLOAD=0                                                           MA 118
      NONET=0                                                           MA 119
      NEAR=-1                                                           MA 120
      IPTFLG=-2                                                         MA 121
      IPTFLQ=-1                                                         MA 122
      IFAR=-1                                                           MA 123
      ZRATI=(1.,0.)                                                     MA 124
      IPED=0                                                            MA 125
      IRNGF=0                                                           MA 126
      NCOUP=0                                                           MA 127
```

```
      ICOUP=0
      IF(ICASX.GT.0)GO TO 14
      FMHZ=CVEL
      NLODF=0
      KSYMP=1
      NRADL=0
      IPERF=0
C
C     MAIN INPUT SECTION - STANDARD READ STATEMENT - JUMPS TO APPRO-
C     PRIATE SECTION FOR SPECIFIC PARAMETER SET UP
C
14    READ(5,136)AIN,ITMP1,ITMP2,ITMP3,ITMP4,TMP1,TMP2,TMP3,TMP4,TMP5,
     1TMP6
      MPCNT=MPCNT+1
      PRINT 137, MPCNT,AIN,ITMP1,ITMP2,ITMP3,ITMP4,TMP1,TMP2,TMP3,TMP4,
     1TMP5,TMP6
      IF (AIN.EQ.ATST(2)) GO TO 16
      IF (AIN.EQ.ATST(3)) GO TO 17
      IF (AIN.EQ.ATST(4)) GO TO 21
      IF (AIN.EQ.ATST(5)) GO TO 24
      IF (AIN.EQ.ATST(6)) GO TO 28
      IF (AIN.EQ.ATST(14)) GO TO 28
      IF (AIN.EQ.ATST(15)) GO TO 31
      IF (AIN.EQ.ATST(18)) GO TO 319
      IF (AIN.EQ.ATST(7)) GO TO 37
      IF (AIN.EQ.ATST(8)) GO TO 32
      IF (AIN.EQ.ATST(17)) GO TO 208
      IF (AIN.EQ.ATST(9)) GO TO 34
      IF (AIN.EQ.ATST(10)) GO TO 36
      IF (AIN.EQ.ATST(16)) GO TO 305
      IF (AIN.EQ.ATST(19)) GO TO 320
      IF (AIN.EQ.ATST(12)) GO TO 1
      IF (AIN.EQ.ATST(20)) GO TO 322
      IF (AIN.EQ.ATST(21)) GO TO 304
      IF (AIN.NE.ATST(13)) GO TO 15
      CALL SECOND(TMP1)
      TMP1=TMP1-EXTIM
      PRINT 201,TMP1
      STOP
15    PRINT 138
      STOP
C
C     FREQUENCY PARAMETERS
C
16    IFRQ=ITMP1
      IF(ICASX.EQ.0)GO TO 8
      PRINT 303,AIN
      STOP
8     NFRQ=ITMP2
      IF (NFRQ.EQ.0) NFRQ=1
      FMHZ=TMP1
      DELFRQ=TMP2
      IF(IPED.EQ.1)ZPNORM=0.
      IGO=1
      IFLOW=1
      GO TO 14
C
C     MATRIX INTEGRATION LIMIT
C
305   RKH=TMP1
      IF(IGO.GT.2)IGO=2
      IFLOW=1
      GO TO 14
C
```

```
C     EXTENDED THIN WIRE KERNEL OPTION
C
  320 IEXK=1
      IF(ITMP1.EQ.-1)IEXK=0
      IF(IGO.GT.2)IGO=2
      IFLOW=1
      GO TO 14
C
C     MAXIMUM COUPLING BETWEEN ANTENNAS
C
  304 IF(IFLOW.NE.2)NCOUP=0
      ICOUP=0
      IFLOW=2
      IF(ITMP2.EQ.0)GO TO 14
      NCOUP=NCOUP+1
      IF(NCOUP.GT.5)GO TO 312
      NCTAG(NCOUP)=ITMP1
      NCSEG(NCOUP)=ITMP2
      IF(ITMP4.EQ.0)GO TO 14
      NCOUP=NCOUP+1
      IF(NCOUP.GT.5)GO TO 312
      NCTAG(NCOUP)=ITMP3
      NCSEG(NCOUP)=ITMP4
      GO TO 14
  312 PRINT 313
      STOP
C
C     LOADING PARAMETERS
C
   17 IF (IFLOW.EQ.3) GO TO 18
      NLOAD=0
      IFLOW=3
      IF (IGO.GT.2) IGO=2
      IF (ITMP1.EQ.(-1)) GO TO 14
   18 NLOAD=NLOAD+1
      IF (NLOAD.LE.LOADMX) GO TO 19
      PRINT 139
      STOP
   19 LDTYP(NLOAD)=ITMP1
      LDTAG(NLOAD)=ITMP2
      IF (ITMP4.EQ.0) ITMP4=ITMP3
      LDTAGF(NLOAD)=ITMP3
      LDTAGT(NLOAD)=ITMP4
      IF (ITMP4.GE.ITMP3) GO TO 20
      PRINT 140, NLOAD,ITMP3,ITMP4
      STOP
   20 ZLR(NLOAD)=TMP1
      ZLI(NLOAD)=TMP2
      ZLC(NLOAD)=TMP3
      GO TO 14
C
C     GROUND PARAMETERS UNDER THE ANTENNA
C
   21 IFLOW=4
      IF(ICASX.EQ.0)GO TO 10
      PRINT 303,AIN
      STOP
   10 IF (IGO.GT.2) IGO=2
      IF (ITMP1.NE.(-1)) GO TO 22
      KSYMP=1
      NRADL=0
      IPERF=0
      GO TO 14
   22 IPERF=ITMP1
```

```
      NRADL=ITMP2
      KSYMP=2
      EPSR=TMP1
      SIG=TMP2
      IF (NRADL.EQ.0) GO TO 23
      IF(IPERF.NE.2)GO TO 314
      PRINT 390
      STOP
314   SCRWLT=TMP3
      SCRWRT=TMP4
      GO TO 14
23    EPSR2=TMP3
      SIG2=TMP4
      CLT=TMP5
      CHT=TMP6
      GO TO 14
C
C     EXCITATION PARAMETERS
C
24    IF (IFLOW.EQ.5) GO TO 25
      NSANT=0
      NVQD=0
      IPED=0
      IFLOW=5
      IF (IGO.GT.3) IGO=3
25    MASYM=ITMP4/10
      IF (ITMP1.GT.0.AND.ITMP1.NE.5) GO TO 27
      IXTYP=ITMP1
      NTSOL=0
      IF(IXTYP.EQ.0)GO TO 205
      NVQD=NVQD+1
      IF(NVQD.GT.NSMAX)GO TO 206
      IVQD(NVQD)=ISEGNO(ITMP2,ITMP3)
      VQD(NVQD)=CMPLX(TMP1,TMP2)
      IF(CABS(VQD(NVQD)).LT.1.E-20)VQD(NVQD)=(1.,0.)
      GO TO 207
205   NSANT=NSANT+1
      IF (NSANT.LE.NSMAX) GO TO 26
206   PRINT 141
      STOP
26    ISANT(NSANT)=ISEGNO(ITMP2,ITMP3)
      VSANT(NSANT)=CMPLX(TMP1,TMP2)
      IF (CABS(VSANT(NSANT)).LT.1.E-20) VSANT(NSANT)=(1.,0.)
207   IPED=ITMP4-MASYM*10
      ZPNORM=TMP3
      IF (IPED.EQ.1.AND.ZPNORM.GT.0) IPED=2
      GO TO 14
27    IF (IXTYP.EQ.0.OR.IXTYP.EQ.5) NTSOL=0
      IXTYP=ITMP1
      NTHI=ITMP2
      NPHI=ITMP3
      XPR1=TMP1
      XPR2=TMP2
      XPR3=TMP3
      XPR4=TMP4
      XPR5=TMP5
      XPR6=TMP6
      NSANT=0
      NVQD=0
      THETIS=XPR1
      PHISS=XPR2
      GO TO 14
C
C     NETWORK PARAMETERS
```

```
C
 28     IF (IFLOW.EQ.6) GO TO 29
        NONET=0
        NTSOL=0
        IFLOW=6
        IF (IGO.GT.3) IGO=3
        IF (ITMP2.EQ.(-1)) GO TO 14
 29     NONET=NONET+1
        IF (NONET.LE.NETMX) GO TO 30
        PRINT 142
        STOP
 30     NTYP(NONET)=2
        IF (AIN.EQ.ATST(6)) NTYP(NONET)=1
        ISEG1(NONET)=ISEGNO(ITMP1,ITMP2)
        ISEG2(NONET)=ISEGNO(ITMP3,ITMP4)
        X11R(NONET)=TMP1
        X11I(NONET)=TMP2
        X12R(NONET)=TMP3
        X12I(NONET)=TMP4
        X22R(NONET)=TMP5
        X22I(NONET)=TMP6
        IF (NTYP(NONET).EQ.1.OR.TMP1.GT.0.) GO TO 14
        NTYP(NONET)=3
        X11R(NONET)=-TMP1
        GO TO 14
C
C       PRINT CONTROL FOR CURRENT
C
 31     IPTFLG=ITMP1
        IPTAG=ITMP2
        IPTAGF=ITMP3
        IPTAGT=ITMP4
        IF(ITMP3.EQ.0.AND.IPTFLG.NE.-1)IPTFLG=-2
        IF (ITMP4.EQ.0) IPTAGT=IPTAGF
        GO TO 14
C
C       PRINT CONTROL FOR CHARGE
C
 319    IPTFLQ=ITMP1
        IPTAQ=ITMP2
        IPTAQF=ITMP3
        IPTAQT=ITMP4
        IF(ITMP3.EQ.0.AND.IPTFLQ.NE.-1)IPTFLQ=-2
        IF(ITMP4.EQ.0)IPTAQT=IPTAQF
        GO TO 14
C
C       NEAR FIELD CALCULATION PARAMETERS
C
 208    NFEH=1
        GO TO 209
 32     NFEH=0
 209    IF (.NOT.(IFLOW.EQ.8.AND.NFRQ.NE.1)) GO TO 33
        PRINT 143
 33     NEAR=ITMP1
        NRX=ITMP2
        NRY=ITMP3
        NRZ=ITMP4
        XNR=TMP1
        YNR=TMP2
        ZNR=TMP3
        DXNR=TMP4
        DYNR=TMP5
        DZNR=TMP6
        IFLOW=8
```

```
      IF (NFRQ.NE.1) GO TO 14
      GO TO (41,46,53,71,72), IGO
C
C     GROUND REPRESENTATION
C
34    EPSR2=TMP1
      SIG2=TMP2
      CLT=TMP3
      CHT=TMP4
      IFLOW=9
      GO TO 14
C
C     STANDARD OBSERVATION ANGLE PARAMETERS
C
36    IFAR=ITMP1
      NTH=ITMP2
      NPH=ITMP3
      IF (NTH.EQ.0) NTH=1
      IF (NPH.EQ.0) NPH=1
      IPD=ITMP4/10
      IAVP=ITMP4-IPD*10
      INOR=IPD/10
      IPD=IPD-INOR*10
      IAX=INOR/10
      INOR=INOR-IAX*10
      IF (IAX.NE.0) IAX=1
      IF (IPD.NE.0) IPD=1
      IF (NTH.LT.2.OR.NPH.LT.2) IAVP=0
      IF (IFAR.EQ.1) IAVP=0
      THETS=TMP1
      PHIS=TMP2
      DTH=TMP3
      DPH=TMP4
      RFLD=TMP5
      GNOR=TMP6
      IFLOW=10
      GO TO (41,46,53,71,78), IGO
C
C     WRITE NUMERICAL GREEN'S FUNCTION TAPE
C
322   IFLOW=12
      IF(ICASX.EQ.0)GO TO 301
      PRINT 302
      STOP
301   IRNGF=IRESRV/2
      GO TO (41,46,52,52,52),IGO
C
C     EXECUTE CARD  -  CALC. INCLUDING RADIATED FIELDS
C
37    IF (IFLOW.EQ.10.AND.ITMP1.EQ.0) GO TO 14
      IF (NFRQ.EQ.1.AND.ITMP1.EQ.0.AND.IFLOW.GT.7) GO TO 14
      IF (ITMP1.NE.0) GO TO 39
      IF (IFLOW.GT.7) GO TO 38
      IFLOW=7
      GO TO 40
38    IFLOW=11
      GO TO 40
39    IFAR=0
      RFLD=0
      IPD=0
      IAVP=0
      INOR=0
      IAX=0
      NTH=91
```

```
      NPH=1
      THETS=0.
      PHIS=0.
      DTH=1.0
      DPH=0.
      IF (ITMP1.EQ.2) PHIS=90.
      IF (ITMP1.NE.3) GO TO 40
      NPH=2
      DPH=90.
40    GO TO (41,46,53,71,78), IGO
C
C     END OF THE MAIN INPUT SECTION
C
C     BEGINNING OF THE FREQUENCY DO LOOP
C
41    MHZ=1
C     CORE ALLOCATION FOR PRIMARY INTERACTON MATRIX.  (A)
      IF(IMAT.EQ.0)CALL FBLOCK(NPEQ,NEQ,IRESRV,IRNGF,IPSYM)
42    IF (MHZ.EQ.1) GO TO 44
      IF (IFRQ.EQ.1) GO TO 43
      FMHZ=FMHZ+DELFRQ
      GO TO 44
43    FMHZ=FMHZ*DELFRQ
44    FR=FMHZ/FMHZS
      WLAM=CVEL/FMHZ
      PRINT 145, FMHZ,WLAM
      PRINT 196,RKH
      IF(IEXK.EQ.1)PRINT 321
C     FREQUENCY SCALING OF GEOMETRIC PARAMETERS
      FMHZS=FMHZ
      IF(N.EQ.0)GO TO 306
      DO 45 I=1,N
      X(I)=X(I)*FR
      Y(I)=Y(I)*FR
      Z(I)=Z(I)*FR
      SI(I)=SI(I)*FR
45    BI(I)=BI(I)*FR
306   IF(M.EQ.0)GO TO 307
      FR2=FR*FR
      J=LD+1
      DO 245 I=1,M
      J=J-1
      X(J)=X(J)*FR
      Y(J)=Y(J)*FR
      Z(J)=Z(J)*FR
245   BI(J)=BI(J)*FR2
307   IGO=2
C     STRUCTURE SEGMENT LOADING
46    PRINT 146
      IF(NLOAD.NE.0) CALL LOAD(LDTYP,LDTAG,LDTAGF,LDTAGT,ZLR,ZLI,ZLC)
      IF(NLOAD.EQ.0.AND.NLODF.EQ.0)PRINT 147
      IF(NLOAD.EQ.0.AND.NLODF.NE.0)PRINT 327
C     GROUND PARAMETER
      PRINT 148
      IF (KSYMP.EQ.1) GO TO 49
      FRATI=(1.,0.)
      IF (IPERF.EQ.1) GO TO 48
      IF(SIG.LT.0.)SIG=-SIG/(59.96*WLAM)
      EPSC=CMPLX(EPSR,-SIG*WLAM*59.96)
      ZRATI=1./CSQRT(EPSC)
      U=ZRATI
      U2=U*U
      IF (NRADL.EQ.0) GO TO 47
      SCRWL=SCRWLT/WLAM
```

```
            SCRWR=SCRWRT/WLAM                                               MA 512
            T1=FJ*2367.067/FLOAT(NRADL)                                     MA 513
            T2=SCRWR*FLOAT(NRADL)                                           MA 514
            PRINT 170, NRADL,SCRWLT,SCRWRT                                  MA 515
            PRINT 149                                                       MA 516
 47         IF(IPERF.EQ.2)GO TO 328                                         MA 517
            PRINT 391                                                       MA 518
            GO TO 329                                                       MA 519
 328        IF(NXA(1).EQ.0)READ(21)AR1,AR2,AR3,EPSCF,DXA,DYA,XSA,YSA,NXA,NYA MA 520
            FRATI=(EPSC-1.)/(EPSC+1.)                                       MA 521
            IF(CABS((EPSCF-EPSC)/EPSC).LT.1.E-3)GO TO 330                   MA 522
            PRINT 393,EPSCF,EPSC                                            MA 523
            STOP                                                            MA 524
 330        PRINT 392                                                       MA 525
 329        PRINT 150, EPSR,SIG,EPSC                                        MA 526
            GO TO 50                                                        MA 527
 48         PRINT 151                                                       MA 528
            GO TO 50                                                        MA 529
 49         PRINT 152                                                       MA 530
 50         CONTINUE                                                        MA 531
 C * * * *                                                                  MA 532
 C          FILL AND FACTOR PRIMARY INTERACTION MATRIX                      MA 533
 C                                                                          MA 534
            CALL SECOND (TIM1)                                              MA 535
            IF(ICASX.NE.0)GO TO 324                                         MA 536
            CALL CMSET(NEQ,CM,RKH,IEXK)                                     MA 537
            CALL SECOND (TIM2)                                              MA 538
            TIM=TIM2-TIM1                                                   MA 539
            CALL FACTRS(NPEQ,NEQ,CM,IP,IX,11,12,13,14)                      MA 540
            GO TO 323                                                       MA 541
 C                                                                          MA 542
 C          N.G.F. - FILL B, C, AND D AND FACTOR D-C(INV(A)B)               MA 543
 C                                                                          MA 544
 324        CALL CMNGF(CM(IB11),CM(IC11),CM(ID11),NPBX,NEQ,NEQ2,RKH,IEXK)   MA 545
            CALL SECOND (TIM2)                                              MA 546
            TIM=TIM2-TIM1                                                   MA 547
            CALL FACGF(CM,CM(IB11),CM(IC11),CM(ID11),CM(IX11),IP,IX,NP,N1,MP, MA 548
           1M1,NEQ,NEQ2)                                                    MA 549
 323        CALL SECOND (TIM1)                                              MA 550
            TIM2=TIM1-TIM2                                                  MA 551
            PRINT 153, TIM,TIM2                                             MA 552
            IGO=3                                                           MA 553
            NTSOL=0                                                         MA 554
            IF(IFLOW.NE.12)GO TO 53                                         MA 555
 C          WRITE N.G.F. FILE                                               MA 556
 52         CALL GFOUT                                                      MA 557
            GO TO 14                                                        MA 558
 C                                                                          MA 559
 C          EXCITATION SET UP (RIGHT HAND SIDE, -E INC.)                    MA 560
 C                                                                          MA 561
 53         NTHIC=1                                                         MA 562
            NPHIC=1                                                         MA 563
            INC=1                                                           MA 564
            NPRINT=0                                                        MA 565
 54         IF (IXTYP.EQ.0.OR.IXTYP.EQ.5) GO TO 56                          MA 566
            IF (IPTFLG.LE.0.OR.IXTYP.EQ.4) PRINT 154                        MA 567
            TMP5=TA*XPR5                                                    MA 568
            TMP4=TA*XPR4                                                    MA 569
            IF (IXTYP.NE.4) GO TO 55                                        MA 570
            TMP1=XPR1/WLAM                                                  MA 571
            TMP2=XPR2/WLAM                                                  MA 572
            TMP3=XPR3/WLAM                                                  MA 573
            TMP6=XPR6/(WLAM*WLAM)                                           MA 574
            PRINT 156, XPR1,XPR2,XPR3,XPR4,XPR5,XPR6                        MA 575
```

```
      GO TO 56
 55   TMP1=TA*XPR1
      TMP2=TA*XPR2
      TMP3=TA*XPR3
      TMP6=XPR6
      IF (IPTFLG.LE.0) PRINT 155, XPR1,XPR2,XPR3,HPOL(IXTYP),XPR6
 56   CALL ETMNS (TMP1,TMP2,TMP3,TMP4,TMP5,TMP6,IXTYP,CUR)
C
C     MATRIX SOLVING  (NETWK CALLS SOLVES)
C
      IF (NONET.EQ.0.OR.INC.GT.1) GO TO 60
      PRINT 158
      ITMP3=0
      ITMP1=NTYP(1)
      DO 59 I=1,2
      IF (ITMP1.EQ.3) ITMP1=2
      IF (ITMP1.EQ.2) PRINT 159
      IF (ITMP1.EQ.1) PRINT 160
      DO 58 J=1,NONET
      ITMP2=NTYP(J)
      IF ((ITMP2/ITMP1).EQ.1) GO TO 57
      ITMP3=ITMP2
      GO TO 58
 57   ITMP4=ISEG1(J)
      ITMP5=ISEG2(J)
      IF (ITMP2.GE.2.AND.X11I(J).LE.0.) X11I(J)=WLAM*SQRT((X(ITMP5)-
     1 X(ITMP4))**2+(Y(ITMP5)-Y(ITMP4))**2+(Z(ITMP5)-Z(ITMP4))**2)
      PRINT 157, ITAG(ITMP4),ITMP4,ITAG(ITMP5),ITMP5,X11R(J),X11I(J),
     1X12R(J),X12I(J),X22R(J),X22I(J),PNET(2*ITMP2-1),PNET(2*ITMP2)
 58   CONTINUE
      IF (ITMP3.EQ.0) GO TO 60
      ITMP1=ITMP3
 59   CONTINUE
 60   CONTINUE
      IF (INC.GT.1.AND.IPTFLG.GT.0) NPRINT=1
      CALL NETWK(CM,CM(IB11),CM(IC11),CM(ID11),IP,CUR)
      NTSOL=1
      IF (IPED.EQ.0) GO TO 61
      ITMP1=MHZ+4*(MHZ-1)
      IF (ITMP1.GT.(NORMF-3)) GO TO 61
      FNORM(ITMP1)=REAL(ZPED)
      FNORM(ITMP1+1)=AIMAG(ZPED)
      FNORM(ITMP1+2)=CABS(ZPED)
      FNORM(ITMP1+3)=CANG(ZPED)
      IF (IPED.EQ.2) GO TO 61
      IF (FNORM(ITMP1+2).GT.ZPNORM) ZPNORM=FNORM(ITMP1+2)
 61   CONTINUE
C
C     PRINTING STRUCTURE CURRENTS
C
      IF(N.EQ.0)GO TO 308
      IF (IPTFLG.EQ.(-1)) GO TO 63
      IF (IPTFLG.GT.0) GO TO 62
      PRINT 161
      PRINT 162
      GO TO 63
 62   IF (IPTFLG.EQ.3.OR.INC.GT.1) GO TO 63
      PRINT 163, XPR3,HPOL(IXTYP),XPR6
 63   PLOSS=0.
      ITMP1=0
      JUMP=IPTFLG+1
      DO 69 I=1,N
      CURI=CUR(I)*WLAM
      CMAG=CABS(CURI)
```

```
      PH=CANG(CURI)                                                     MA 640
      IF (NLOAD.EQ.0.AND.NLODF.EQ.0) GO TO 64                           MA 641
      IF (ABS(REAL(ZARRAY(I))).LT.1.E-20) GO TO 64                      MA 642
      PLOSS=PLOSS+.5*CMAG*CMAG*REAL(ZARRAY(I))*SI(I)                    MA 643
64    IF (JUMP) 68,69,65                                                MA 644
65    IF (IPTAG.EQ.0) GO TO 66                                          MA 645
      IF (ITAG(I).NE.IPTAG) GO TO 69                                    MA 646
66    ITMP1=ITMP1+1                                                     MA 647
      IF (ITMP1.LT.IPTAGF.OR.ITMP1.GT.IPTAGT) GO TO 69                  MA 648
      IF (IPTFLG.EQ.0) GO TO 68                                         MA 649
      IF (IPTFLG.LT.2.OR.INC.GT.NORMF) GO TO 67                         MA 650
      FNORM(INC)=CMAG                                                   MA 651
      ISAVE=I                                                           MA 652
67    IF (IPTFLG.NE.3) PRINT 164, XPR1,XPR2,CMAG,PH,I                   MA 653
      GO TO 69                                                          MA 654
68    PRINT 165, I,ITAG(I),X(I),Y(I),Z(I),SI(I),CURI,CMAG,PH            MA 655
69    CONTINUE                                                          MA 656
      IF(IPTFLQ.EQ.(-1))GO TO 308                                       MA 657
      PRINT 315                                                         MA 658
      ITMP1=0                                                           MA 659
      FR=1.E-6/FMHZ                                                     MA 660
      DO 316 I=1,N                                                      MA 661
      IF(IPTFLQ.EQ.(-2))GO TO 318                                       MA 662
      IF(IPTAQ.EQ.0)GO TO 317                                           MA 663
      IF(ITAG(I).NE.IPTAQ)GO TO 316                                     MA 664
317   ITMP1=ITMP1+1                                                     MA 665
      IF(ITMP1.LT.IPTAQF.OR.ITMP1.GT.IPTAQT)GO TO 316                   MA 666
318   CURI=FR*CMPLX(-BII(I),BIR(I))                                     MA 667
      CMAG=CABS(CURI)                                                   MA 668
      PH=CANG(CURI)                                                     MA 669
      PRINT 165,I,ITAG(I),X(I),Y(I),Z(I),SI(I),CURI,CMAG,PH             MA 670
316   CONTINUE                                                          MA 671
308   IF(M.EQ.0)GO TO 310                                               MA 672
      PRINT 197                                                         MA 673
      J=N-2                                                             MA 674
      ITMP1=LD+1                                                        MA 675
      DO 309 I=1,M                                                      MA 676
      J=J+3                                                             MA 677
      ITMP1=ITMP1-1                                                     MA 678
      EX=CUR(J)                                                         MA 679
      EY=CUR(J+1)                                                       MA 680
      EZ=CUR(J+2)                                                       MA 681
      ETH=EX*T1X(ITMP1)+EY*T1Y(ITMP1)+EZ*T1Z(ITMP1)                     MA 682
      EPH=EX*T2X(ITMP1)+EY*T2Y(ITMP1)+EZ*T2Z(ITMP1)                     MA 683
      ETHM=CABS(ETH)                                                    MA 684
      ETHA=CANG(ETH)                                                    MA 685
      EPHM=CABS(EPH)                                                    MA 686
      EPHA=CANG(EPH)                                                    MA 687
309   PRINT 198,I,X(ITMP1),Y(ITMP1),Z(ITMP1),ETHM,ETHA,EPHM,EPHA,EX,EY, MA 688
     1 EZ                                                               MA 689
310   IF (IXTYP.NE.0.AND.IXTYP.NE.5) GO TO 70                           MA 690
      TMP1=PIN-PNLS-PLOSS                                               MA 691
      TMP2=100.*TMP1/PIN                                                MA 692
      PRINT 166, PIN,TMP1,PLOSS,PNLS,TMP2                               MA 693
70    CONTINUE                                                          MA 694
      IGO=4                                                             MA 695
      IF(NCOUP.GT.0)CALL COUPLE(CUR,WLAM)                               MA 696
      IF (IFLOW.NE.7) GO TO 71                                          MA 697
      IF (IXTYP.GT.0.AND.IXTYP.LT.4) GO TO 113                          MA 698
      IF (NFRQ.NE.1) GO TO 120                                          MA 699
      PRINT 135                                                         MA 700
      GO TO 14                                                          MA 701
71    IGO=5                                                             MA 702
C                                                                       MA 703
```

```
C       NEAR FIELD CALCULATION
C
72      IF (NEAR.EQ.(-1)) GO TO 78
        CALL NFPAT
        IF (MHZ.EQ.NFRQ) NEAR=-1
        IF (NFRQ.NE.1) GO TO 78
        PRINT 135
        GO TO 14
C
C       STANDARD FAR FIELD CALCULATION
C
78      IF(IFAR.EQ.-1)GO TO 113
        PINR=PIN
        PNLR=PNLS
        CALL RDPAT
113     IF (IXTYP.EQ.0.OR.IXTYP.GE.4) GO TO 119
        NTHIC=NTHIC+1
        INC=INC+1
        XPR1=XPR1+XPR4
        IF (NTHIC.LE.NTHI) GO TO 54
        NTHIC=1
        XPR1=THETIS
        XPR2=XPR2+XPR5
        NPHIC=NPHIC+1
        IF (NPHIC.LE.NPHI) GO TO 54
        NPHIC=1
        XPR2=PHISS
        IF (IPTFLG.LT.2) GO TO 119
C       NORMALIZED RECEIVING PATTERN PRINTED
        ITMP1=NTHI*NPHI
        IF (ITMP1.LE.NORMF) GO TO 114
        ITMP1=NORMF
        PRINT 181
114     TMP1=FNORM(1)
        DO 115 J=2,ITMP1
        IF (FNORM(J).GT.TMP1) TMP1=FNORM(J)
115     CONTINUE
        PRINT 182, TMP1,XPR3,HPOL(IXTYP),XPR6,ISAVE
        DO 118 J=1,NPHI
        ITMP2=NTHI*(J-1)
        DO 116 I=1,NTHI
        ITMP3=I+ITMP2
        IF (ITMP3.GT.ITMP1) GO TO 117
        TMP2=FNORM(ITMP3)/TMP1
        TMP3=DB20(TMP2)
        PRINT 183, XPR1,XPR2,TMP3,TMP2
        XPR1=XPR1+XPR4
116     CONTINUE
117     XPR1=THETIS
        XPR2=XPR2+XPR5
118     CONTINUE
        XPR2=PHISS
119     IF (MHZ.EQ.NFRQ) IFAR=-1
        IF (NFRQ.NE.1) GO TO 120
        PRINT 135
        GO TO 14
120     MHZ=MHZ+1
        IF (MHZ.LE.NFRQ) GO TO 42
        IF (IPED.EQ.0) GO TO 123
        IF(NVQD.LT.1)GO TO 199
        PRINT 184,IVQD(NVQD),ZPNORM
        GO TO 204
199     PRINT 184, ISANT(NSANT),ZPNORM
204     ITMP1=NFRQ
```

```
      IF (ITMP1.LE.(NORMF/4)) GO TO 121
      ITMP1=NORMF/4
      PRINT 185
121   IF (IFRQ.EQ.0) TMP1=FMHZ-(NFRQ-1)*DELFRQ
      IF (IFRQ.EQ.1) TMP1=FMHZ/(DELFRQ**(NFRQ-1))
      DO 122 I=1,ITMP1
      ITMP2=I+4*(I-1)
      TMP2=FNORM(ITMP2)/ZPNORM
      TMP3=FNORM(ITMP2+1)/ZPNORM
      TMP4=FNORM(ITMP2+2)/ZPNORM
      TMP5=FNORM(ITMP2+3)
      PRINT 186, TMP1,FNORM(ITMP2),FNORM(ITMP2+1),FNORM(ITMP2+2),
     1FNORM(ITMP2+3),TMP2,TMP3,TMP4,TMP5
      IF (IFRQ.EQ.0) TMP1=TMP1+DELFRQ
      IF (IFRQ.EQ.1) TMP1=TMP1*DELFRQ
122   CONTINUE
      PRINT 135
123   CONTINUE
      NFRQ=1
      MHZ=1
      GO TO 14
125   FORMAT (A2,13A6)
126   FORMAT  (1H1)
127   FORMAT (///,33X,36H************************************,//,36X,
     1 31HNUMERICAL ELECTROMAGNETICS CODE,//,33X,
     2 36H************************************)
128   FORMAT (////,37X,24H- - - - COMMENTS - - - -,//)
129   FORMAT (25X,13A6)
130   FORMAT (///,10X,34HINCORRECT LABEL FOR A COMMENT CARD)
135   FORMAT (/////)
136   FORMAT (A2,I3,3I5,6E10.3)
137   FORMAT (1X, 19H***** DATA CARD NO.,I3,3X,A2,1X,I3,3(1X,I5),
     1 6(1X,E12.5))
138   FORMAT (///,10X,45HFAULTY DATA CARD LABEL AFTER GEOMETRY SECTION)
139   FORMAT (///,10X,48HNUMBER OF LOADING CARDS EXCEEDS STORAGE ALLOTTE
     1D)
140   FORMAT (///,10X,31HDATA FAULT ON LOADING CARD NO.=,I5,5X,11HITAG S
     1TEP1=,I5,29H  IS GREATER THAN ITAG STEP2=,I5)
141   FORMAT (///,10X,51HNUMBER OF EXCITATION CARDS EXCEEDS STORAGE ALLO
     1TTED)
142   FORMAT (///,10X,48HNUMBER OF NETWORK CARDS EXCEEDS STORAGE ALLOTTE
     1D)
143   FORMAT(///,10X,79HWHEN MULTIPLE FREQUENCIES ARE REQUESTED, ONLY ON
     1E NEAR FIELD CARD CAN BE USED -,/,10X,22HLAST CARD READ IS USED)
145   FORMAT (////,33X,33H- - - - - - FREQUENCY - - - - - -,//,36X,10HFR
     1EQUENCY=,E11.4,4H MHZ,/,36X,11HWAVELENGTH=,E11.4,7H METERS)
146   FORMAT (///,30X,40H - - - STRUCTURE IMPEDANCE LOADING - - -)
147   FORMAT (/ ,35X,28HTHIS STRUCTURE IS NOT LOADED)
148   FORMAT (///,34X,31H- - - ANTENNA ENVIRONMENT - - -,/)
149   FORMAT (40X,21HMEDIUM UNDER SCREEN -)
150   FORMAT (40X,27HRELATIVE DIELECTRIC CONST.=,F7.3,/,40X,13HCONDUCTIV
     1ITY=,E10.3,11H MHOS/METER,/,40X,28HCOMPLEX DIELECTRIC CONSTANT=,
     12E12.5)
151   FORMAT (  42X,14HPERFECT GROUND)
152   FORMAT (  44X,10HFREE SPACE)
153   FORMAT (///,32X,25H- - - MATRIX TIMING - - -,//,24X,5HFILL=,F9.3,
     115H SEC.,  FACTOR=,F9.3,5H SEC.)
154   FORMAT (///,40X,22H- - - EXCITATION - - -)
155   FORMAT (/,4X,10HPLANE WAVE,4X,6HTHETA=,F7.2,11H DEG,  PHI=,F7.2,
     1 11H DEG,  ETA=,F7.2,13H DEG,  TYPE -,A6,15H=  AXIAL RATIO=,F6.3)
156   FORMAT (/,31X,17HPOSITION (METERS),14X,18HORIENTATION (DEG)=/,28X,
     11HX,12X,1HY,12X,1HZ,10X,5HALPHA,5X,4HBETA,4X,13HDIPOLE MOMENT,//
     2 ,4X,14HCURRENT SOURCE,1X,3(3X,F10.5),1X,2(3X,F7.2),4X,F8.3)
157   FORMAT (4X,4(I5,1X),6(3X,E11.4),3X,A6,A2)
```

```
158   FORMAT (///,44X,24H- - - NETWORK DATA - - -)                          MA 832
159   FORMAT (/,6X,18H- FROM -    - TO -,11X,17HTRANSMISSION LINE,15X,36    MA 833
     1H-  -  SHUNT ADMITTANCES (MHOS)  -  -,14X,4HLINE,/,6X,21HTAG  SEG.    MA 834
     2  TAG  SEG.,6X,9HIMPEDANCE,6X,6HLENGTH,12X,11H- END ONE -,17X,11H     MA 835
     3- END TWO -,12X,4HTYPE,/    ,6X,21HNO.    NO.    NO.    NO.,9X,4HOHMS MA 836
     4,8X,6HMETERS,9X, 4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.)            MA 837
160   FORMAT (/,6X,8H- FROM -,4X,6H- TO -,26X,45H-  -  ADMITTANCE MATRIX    MA 838
     1 ELEMENTS (MHOS)  -  -,/     ,6X,21HTAG  SEG.   TAG  SEG.,13X,9H(ON   MA 839
     2E,ONE),19X,    9H(ONE,TWO),19X,9H(TWO,TWO),/ ,6X,21HNO.   NO.    NO   MA 840
     3.   NO.,8X,4HREAL,10X,5HIMAG.,9X,4HREAL,10X,5HIMAG.,9X,4HREAL,        MA 841
     4 10X,5HIMAG.)                                                         MA 842
161   FORMAT (///,29X,33H- - - CURRENTS AND LOCATION - - -,//,33X,24HDIS    MA 843
     1TANCES IN WAVELENGTHS)                                                MA 844
162   FORMAT (  //,2X,4HSEG.,2X,3HTAG,4X,21HCOORD. OF SEG. CENTER,5X,       MA 845
     1 4HSEG.,12X,26H- - - CURRENT (AMPS) - - -,/,2X,3HNO.,3X,3HNO.,        MA 846
     2 5X,1HX,8X,1HY,8X,1HZ,6X,6HLENGTH,5X,4HREAL,8X,5HIMAG.,7X,4HMAG.,     MA 847
     3 8X,5HPHASE)                                                          MA 848
163   FORMAT (///,33X,40H- - - RECEIVING PATTERN PARAMETERS - - -,/  ,43    MA 849
     1X,4HETA=,F7.2,8H DEGREES,/,43X,6HTYPE -,A6,/,43X,12HAXIAL RATIO=,     MA 850
     2 F6.3,//   ,11X,5HTHETA,6X,3HPHI,10X,13H-  CURRENT  -,9X,3HSEG,/      MA 851
     3,11X,5H(DEG),5X,5H(DEG),7X,9HMAGNITUDE,4X,5HPHASE,6X,3HNO.,/)         MA 852
164   FORMAT (10X,2(F7.2,3X),1X,E11.4,3X,F7.2,4X,I5)                        MA 853
165   FORMAT (1X,2I5,3F9.4,F9.5,1X,3E12.4,F9.3)                             MA 854
166   FORMAT (///,40X,24H- - - POWER BUDGET - - -,//    ,43X,15HINPUT PO    MA 855
     1WER   =,E11.4,6H WATTS,/ ,43X,15HRADIATED POWER=,E11.4,6H WATTS,/     MA 856
     2 ,43X,15HSTRUCTURE LOSS=,E11.4,6H WATTS,/ ,43X,15HNETWORK LOSS  =,    MA 857
     3  E11.4,6H WATTS,/,43X,15HEFFICIENCY     =, F7.2,8H PERCENT)          MA 858
170   FORMAT (40X,25HRADIAL WIRE GROUND SCREEN,/,40X,   I5,6H WIRES,/,40    MA 859
     1X,12HWIRE LENGTH=,F8.2,7H METERS,/,40X,12HWIRE RADIUS=,E10.3,7H ME    MA 860
     2TERS)                                                                 MA 861
181   FORMAT (///,4X,51HRECEIVING PATTERN STORAGE TOO SMALL,ARRAY TRUNCA    MA 862
     1TED)                                                                  MA 863
182   FORMAT (///,32X,40H- - - NORMALIZED RECEIVING PATTERN - - -,/,41X,    MA 864
     1 21HNORMALIZATION FACTOR=,E11.4,/,41X,4HETA=,F7.2,8H DEGREES,/,41X    MA 865
     2,6HTYPE -,A6,/,41X,12HAXIAL RATIO=,F6.3,/,41X,12HSEGMENT NO.=,I5,/    MA 866
     3/,21X,5HTHETA,6X,3HPHI,9X,13H-  PATTERN  -,/,21X,5H(DEG),5X,5H(DEG    MA 867
     4),8X,2HDB,8X,9HMAGNITUDE,/)                                           MA 868
183   FORMAT (20X,2(F7.2,3X),1X,F7.2,4X,E11.4)                              MA 869
184   FORMAT (///,36X,32H- - - INPUT IMPEDANCE DATA - - -,/    ,45X,18HSO   MA 870
     1URCE SEGMENT NO.,I4,/   ,45X,21HNORMALIZATION FACTOR=,E12.5,//        MA 871
     2,7X,5HFREQ.,13X,34H-   -  UNNORMALIZED IMPEDANCE  -  -,21X,   32H-    MA 872
     3 -  NORMALIZED IMPEDANCE  -  -,/      ,19X,10HRESISTANCE,4X,9HREACTA  MA 873
     4NCE,6X,9HMAGNITUDE,4X,5HPHASE,7X,10HRESISTANCE,4X,9HREACTANCE,6X,     MA 874
     5 9HMAGNITUDE,4X,5HPHASE,/     ,8X,3HMHZ,11X,4HOHMS,10X,4HOHMS,11X,    MA 875
     6 4HOHMS,5X,7HDEGREES,47X,7HDEGREES,/)                                 MA 876
185   FORMAT (///,4X,62HSTORAGE FOR IMPEDANCE NORMALIZATION TOO SMALL, A    MA 877
     1RRAY TRUNCATED)                                                       MA 878
186   FORMAT (3X,F9.3,2X,2(2X,E12.5),3X,E12.5,2X,F7.2,2X,2(2X,E12.5),3X,    MA 879
     1 E12.5,2X,F7.2)                                                       MA 880
196   FORMAT(    ////,20X,55HAPPROXIMATE INTEGRATION EMPLOYED FOR SEGMENT   MA 881
     1S MORE THAN,F8.3,18H WAVELENGTHS APART)                               MA 882
197   FORMAT(    ////,41X,38H- - - - SURFACE PATCH CURRENTS - - - -,//,     MA 883
     1 50X,23HDISTANCE IN WAVELENGTHS,/,50X,21HCURRENT IN AMPS/METER,       MA 884
     1 //,28X,26H- - SURFACE COMPONENTS - -,19X,34H- - - RECTANGULAR COM    MA 885
     1PONENTS - - -,/,6X,12HPATCH CENTER,6X,16HTANGENT VECTOR 1,3X,         MA 886
     116HTANGENT VECTOR 2,11X,1HX,19X,1HY,19X,1HZ,/,5X,1HX,6X,1HY,6X,       MA 887
     11HZ,5X,4HMAG.,7X,5HPHASE,3X,4HMAG.,7X,5HPHASE,3(4X,4HREAL,6X,         MA 888
     1 6HIMAG. ))                                                           MA 889
198   FORMAT(1X,I4,/,1X,3F7.3,2(E11.4,F8.2),6E10.2)                         MA 890
201   FORMAT(/,11H RUN TIME =,F10.3)                                        MA 891
315   FORMAT(///,34X,28H- - - CHARGE DENSITIES - - -,//,36X,                MA 892
     1 24HDISTANCES IN WAVELENGTHS,///,2X,4HSEG.,2X,3HTAG,4X,               MA 893
     2 21HCOORD. OF SEG. CENTER,5X,4HSEG.,10X,                              MA 894
     3 31HCHARGE DENSITY (COULOMBS/METER),/,2X,3HNO.,3X,3HNO.,5X,1HX,8X,    MA 895
```

```
     4 1HY,8X,1HZ,6X,6HLENGTH,5X,4HREAL,8X,5HIMAG.,7X,4HMAG.,8X,5HPHASE) MA 896
321   FORMAT( /,20X,42HTHE EXTENDED THIN WIRE KERNEL WILL BE USED)       MA 897
303   FORMAT(/,9H ERROR - ,A2,32H CARD IS NOT ALLOWED WITH N.G.F.)       MA 898
327   FORMAT(/,35X,31H LOADING ONLY IN N.G.F. SECTION)                   MA 899
302   FORMAT(48H ERROR - N.G.F. IN USE.  CANNOT WRITE NEW N.G.F.)        MA 900
313   FORMAT(/,62H NUMBER OF SEGMENTS IN COUPLING CALCULATION (CP) EXCEE MA 901
     1DS LIMIT)                                                          MA 902
390   FORMAT(78H RADIAL WIRE G. S. APPROXIMATION MAY NOT BE USED WITH SO MA 903
     1MMERFELD GROUND OPTION)                                            MA 904
391   FORMAT(40X,52HFINITE GROUND.  REFLECTION COEFFICIENT APPROXIMATION MA 905
     1)                                                                  MA 906
392   FORMAT(40X,35HFINITE GROUND.  SOMMERFELD SOLUTION)                 MA 907
393   FORMAT(/,29H ERROR IN GROUND PARAMETERS -,/,41H COMPLEX DIELECTRIC MA 908
     1 CONSTANT FROM FILE IS,2E12.5,/,32X,9HREQUESTED,2E12.5)            MA 909
      END                                                                MA 910-
```

ARC

PURPOSE

To fill COMMON/DATA/ with segment coordinates for a circular arc of segments.

METHOD

The formal parameters specify the number of segments, radius of the arc, starting angle, final angle and wire radius. Segment coordinates are computed for the arc in the x, z plane with a left hand rotation about the y axis.

SYMBOL DICTIONARY

ANG	= angle of point on the arc (radians, zero on x axis)
ANG1	= angle at first end
ANG2	= angle at second end
DANG	= angle covered by each segment
IST	= number of initial segment
ITG	= tag number assigned to each segment
NS	= number of segments
RAD	= wire radius
RADA	= arc radius
TA	= $\pi/180$
XS1	= x coordinate of first end of segment
XS2	= x coordinate of second end of segment
ZS1	= z coordinate of first end of segment
ZS2	= z coordinate of second end of segment

CONSTANTS

.01745329252	= $\pi/180$
360.00001	= test for angle greater than 360 degrees

ARC

```
      SUBROUTINE ARC (ITG,NS,RADA,ANG1,ANG2,RAD)
C
C     ARC GENERATES SEGMENT GEOMETRY DATA FOR AN ARC OF NS SEGMENTS
C
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      DIMENSION X2(1), Y2(1), Z2(1)
      EQUIVALENCE (X2,SI), (Y2,ALP), (Z2,BET)
      DATA TA/.01745329252/
      IST=N+1
      N=N+NS
      NP=N
      MP=M
      IPSYM=0
      IF (NS.LT.1) RETURN
      IF (ABS(ANG2-ANG1).LT.360.00001) GO TO 1
      PRINT 3
      STOP
    1 ANG=ANG1*TA
      DANG=(ANG2-ANG1)*TA/NS
      XS1=RADA*COS(ANG)
      ZS1=RADA*SIN(ANG)
      DO 2 I=IST,N
      ANG=ANG+DANG
      XS2=RADA*COS(ANG)
      ZS2=RADA*SIN(ANG)
      X(I)=XS1
      Y(I)=0.
      Z(I)=ZS1
      X2(I)=XS2
      Y2(I)=0.
      Z2(I)=ZS2
      XS1=XS2
      ZS1=ZS2
      BI(I)=RAD
    2 ITAG(I)=ITG
      RETURN
C
    3 FORMAT (40H ERROR -- ARC ANGLE EXCEEDS 360. DEGREES)
      END
```

ATGN2

PURPOSE

To return zero when both arguments of a two-argument arctangent function are zero. (Most standard arctangent functions give an error return when both arguments are zero.)

METHOD

System function ATAN2 is used except when both arguments are zero, in which case the value zero is returned. The value returned is the angle (in radians) whose sine is X and cosine is Y.

SYMBOL DICTIONARY

X = first argument

Y = second argument

CODE LISTING

```
      FUNCTION ATGN2 (X,Y)                                             AT   1
C                                                                      AT   2
C     ATGN2 IS ARCTANGENT FUNCTION MODIFIED TO RETURN 0. WHEN X=Y=0.   AT   3
C                                                                      AT   4
      IF (X) 3,1,3                                                     AT   5
    1 IF (Y) 3,2,3                                                     AT   6
    2 ATGN2=0.                                                         AT   7
      RETURN                                                           AT   8
    3 ATGN2=ATAN2(X,Y)                                                 AT   9
      RETURN                                                           AT  10
      END                                                              AT  11-
```

BLCKOT

PURPOSE

To control the writing and reading of matrix blocks on files for the out-of-core matrix solution. The routine also checks for the end-of-file condition during reading.

METHOD

The routine uses a binary read and write with implied DO loops for reading and writing variable length strings into and out of various core locations. The end-of-file condition is checked by a call to function ENF. If an unexpected end of file is detected (governed by NEOF) the program stops.

CODING

BL9 - BL12 Write a record on file NUNIT.

BL14 - BL20 Read NBLKS records from NUNIT, and check for end of file.

BL21 - BL24 Code if end of file detected.

SYMBOL DICTIONARY

AR	= matrix array
ENF	= external function (checks end-of-file condition)
I	= DO loop index
I1, I2	= implied DO loop limits, inclusive matrix locations written from or read into
J	= implied DO index
NBLKS	= number of records to be read
NEOF	= EOF check flag, also used to trace the call to BLCKOT
NUNIT	= file number

CONSTANT

777 = NEOF when EOF is expected by calling program

```
      SUBROUTINE BLCKOT (AR,NUNIT,IX1,IX2,NBLKS,NEOF)
C
C     BLCKOT CONTROLS THE READING AND WRITING OF MATRIX BLOCKS ON FILES
C     FOR THE OUT-OF-CORE MATRIX SOLUTION.
C
      LOGICAL ENF
      COMPLEX AR
      DIMENSION AR(1)
      I1=(IX1+1)/2
      I2=(IX2+1)/2
1     WRITE (NUNIT) (AR(J),J=I1,I2)
      RETURN
      ENTRY BLCKIN
      I1=(IX1+1)/2
      I2=(IX2+1)/2
      DO 2 I=1,NBLKS
      READ (NUNIT) (AR(J),J=I1,I2)
      IF (ENF(NUNIT)) GO TO 3
2     CONTINUE
      RETURN
3     PRINT 4, NUNIT,NBLKS,NEOF
      IF (NEOF.NE.777) STOP
      NEOF=0
      RETURN
C
4     FORMAT (13H  EOF ON UNIT,I3,9H  NBLKS= ,I3,8H  NEOF= ,I5)
      END
```

CABC

PURPOSE

To compute the coefficients in the current function on each segment, given the basis function amplitudes. Surface current components are also computed.

METHOD

The total current on segment i is

$$I_i(s) = A_i + B_i \sin [k(s - s_i)] + C_i \cos [k(s - s_i)] ,$$

where s is distance along the wire, and $s = s_i$ at the center of segment i. The coefficients A_i, B_i, and C_i are the sums of the corresponding coefficients in the portion of each basis function that extends onto segment i.

CODING

CB35	Call to TBF computes components of basis function I.
CB36 - CB43	The basis function components are multiplied by the basis function amplitude from array CURX and summed for each segment.
CB45 - CB63	For a current slope discontinuity source, the special basis function with discontinuous slope, from which the exciting electric field was computed, is recomputed and added to the current coefficients. The call to TBF, with the second argument zero and ICON1(I) temporarily zero, computes a basis function going to zero with non-zero derivative at end one of segment I.
CB64 - CB65	Total current at the center of each segment is computed and stored in place of the basis function amplitudes.
CB68 - CB79	The $\hat{t}_1$ and $\hat{t}_2$ components of surface current for each patch are expanded to x, y, and z components.

SYMBOL DICTIONARY

AR, AI = real and imaginary parts of the basis function amplitude

CCJ, CCX = $-j/60$

CS1, CS2 = $\hat{t}_1$ and $\hat{t}_2$ components of surface current on a patch

CURD = amplitude of the special basis function for a current slope discontinuity source

CURX = input array of basis function amplitudes that are replaced by values of current at segment centers

J = number of a segment onto which a basis function extends

JC01, JC02 = array locations of the $\hat{t}_1$ and $\hat{t}_2$ surface current components for a patch

JX = DO loop index; temporary storage of connection number

K = array location for patch geometry data

SH = (half segment length)/λ

TP = 2π

```
      SUBROUTINE CABC (CURX)
C
C     CABC COMPUTES COEFFICIENTS OF THE CONSTANT (A), SINE (B), AND
C     COSINE (C) TERMS IN THE CURRENT INTERPOLATION FUNCTIONS FOR THE
C     CURRENT VECTOR CUR.
C
      COMPLEX CUR,CURX,VQDS,CURD,CCJ,VSANT,VQD,CS1,CS2
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),CII(300
     1),CUR(900)
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP
     1CON(10),NPCON
      COMMON /VSORC/ VQD(30),VSANT(30),VQDS(30),IVQD(30),ISANT(30),IQDS(
     130),NVQD,NSANT,NQDS
      COMMON /ANGL/ SALP(300)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      DIMENSION CURX(1), CCJX(2)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      EQUIVALENCE (CCJ,CCJX)
      DATA TP/6.283185308/,CCJX/0.,-0.01666666667/
      IF (N.EQ.0) GO TO 6
      DO 1 I=1,N
      AIR(I)=0.
      AII(I)=0.
      BIR(I)=0.
      BII(I)=0.
      CIR(I)=0.
    1 CII(I)=0.
      DO 2 I=1,N
      AR=REAL(CURX(I))
      AI=AIMAG(CURX(I))
      CALL TBF (I,1)
      DO 2 JX=1,JSNO
      J=JCO(JX)
      AIR(J)=AIR(J)+AX(JX)*AR
      AII(J)=AII(J)+AX(JX)*AI
      BIR(J)=BIR(J)+BX(JX)*AR
      BII(J)=BII(J)+BX(JX)*AI
      CIR(J)=CIR(J)+CX(JX)*AR
    2 CII(J)=CII(J)+CX(JX)*AI
      IF (NQDS.EQ.0) GO TO 4
      DO 3 IS=1,NQDS
      I=IQDS(IS)
      JX=ICON1(I)
      ICON1(I)=0
      CALL TBF (I,0)
      ICON1(I)=JX
      SH=SI(I)*.5
      CURD=CCJ*VQDS(IS)/((ALOG(2.*SH/BI(I))-1.)*(BX(JSNO)*COS(TP*SH)+CX(
     1JSNO)*SIN(TP*SH))*WLAM)
      AR=REAL(CURD)
      AI=AIMAG(CURD)
      DO 3 JX=1,JSNO
      J=JCO(JX)
      AIR(J)=AIR(J)+AX(JX)*AR
      AII(J)=AII(J)+AX(JX)*AI
      BIR(J)=BIR(J)+BX(JX)*AR
      BII(J)=BII(J)+BX(JX)*AI
      CIR(J)=CIR(J)+CX(JX)*AR
    3 CII(J)=CII(J)+CX(JX)*AI
    4 DO 5 I=1,N
```

```
5     CURX(I)=CMPLX(AIR(I)+CIR(I),AII(I)+CII(I))
6     IF (M.EQ.0) RETURN
C     CONVERT SURFACE CURRENTS FROM T1,T2 COMPONENTS TO X,Y,Z COMPONENTS
      K=LD-M
      JCO1=N+2*M+1
      JCO2=JCO1+M
      DO 7 I=1,M
      K=K+1
      JCO1=JCO1-2
      JCO2=JCO2-3
      CS1=CURX(JCO1)
      CS2=CURX(JCO1+1)
      CURX(JCO2)=CS1*T1X(K)+CS2*T2X(K)
      CURX(JCO2+1)=CS1*T1Y(K)+CS2*T2Y(K)
7     CURX(JCO2+2)=CS1*T1Z(K)+CS2*T2Z(K)
      RETURN
      END
```

CANG

PURPOSE

To calculate the phase angle of a complex number in degrees.

METHOD

$z = x + jy$

$\phi = [\arctan (y/x)]\ 57.29577951$

SYMBOL DICTIONARY

AIMAG = external routine (imaginary part of complex number)

ATGN2 = external routine (arctan for all quadrants)

CANG = ϕ

REAL = external routine (real part of a complex number)

Z = input complex quantity

CONSTANT

57.29577951 conversion from radians to degrees

CODE LISTING

```
      FUNCTION CANG (Z)                                                 CA   1
C                                                                       CA   2
      CANG RETURNS THE PHASE ANGLE OF A COMPLEX NUMBER IN DEGREES.      CA   3
C                                                                       CA   4
      COMPLEX Z                                                         CA   5
      CANG=ATGN2(AIMAG(Z),REAL(Z))*57.29577951                          CA   6
      RETURN                                                            CA   7
      END                                                               CA   8-
```

CMNGF

PURPOSE

To compute and store the matrices B, C and D for the NGF solution.

METHOD

The structure of matrices B, C and D is described in Section VI. The coding to fill these matrices is involved due to their complex structure, as shown in Figure 12 of Section VI. The complexity is increased by the need to divide the matrices into blocks of rows when they are stored on files (see Section VII).

Much of the coding in CMNGF has to do with connections between new and NGF segments and patches. When a new segment or patch connects to a NGF segment the basis function associated with the NGF segment is modified due to the new junction condition. The amplitude of the modified basis function is a new unknown associated with the B' and D' sections of the matrix. The modified basis function may extend onto other NGF segments that may or may not connect directly to new segments. Also, the basis function of the new segment extends onto the NGF segment to which it connects. Hence fields must be computed for the currents on some NGF segments as well as all new segments.

Comments in the code should be of some help in understanding the procedure. The notation D(WS) in the comments corresponds to D_{sw} in Figure 12. Some parts of the code are explained below.

CG61 - CG70 TRIO computes the components of all basis functions on segment J, where J is a new segment, and stores the coefficients in COMMON/SEGJ/. The array JCO contains the basis-function numbers which ordinarily are the matrix columns associated with the basis functions. If the basis function is for a new segment then JCO is set at CG66 to the column relative to the beginning of the matrix B. If the basis function is for a NGF segment modified by the connection, then JCO is set at CG68 to the column in B'_{ww} relative to the beginning of B. Thus the calls to CMWW and CMWS may store contributions in B'_{ww} and B'_{sw} as well as B_{ww} and B_{sw}.

CG90 - CG108 In this section the fields are evaluated for NGF segments that connect to new segments or patches. TRIO finds all basis functions that contribute to the current on the segment. For a component of a new basis function IR is set to the column in B_{ww} at CG95. For a component of a modified basis function IR is set to the column in B'_{ww}, relative to the start of B, at CG99. If the basis function component is for a NGF basis function that has not been modified the test at CG98 skips to the end of the loop. The arrays in COMMON/SEGJ/ are adjusted from CG101 to CG104 so that CMWW and CMWS will store the matrix element contributions in the correct locations.

CG109 - CG119 If a NGF segment connects to a new segment on one end and to a NGF patch on the opposite end the modified basis function extends onto the patch as a singular component of the patch current. The field due to this component on the patch is added to the matrix element of the modified basis function at CG119.

CG122-CG136 This is similar to CG90 to CG108, but evaluates fields of NGF segments that get contributions from modified basis functions, but do not connect directly to new segments. TBF is called, rather than TRIO to compute modified basis function J on all segments on which it exists. New segments and NGF segments for which contributions have already been evaluated are skipped at CG133 and CG134.

CG165 - CG263 Filling C and D is similar to that for B but fields must be evaluated for all NGF segments and patches as well as new segments and patches.

SYMBOL DICTIONARY

CB = array for matrix B

CC = array for matrix C

CD = array for matrix D

IEXKX = flag to select extended thin-wire kernel

MIEQ = number of patch equations in NGF

MEQ = total number of patch equations

NB = row dimension of CB. CB will contain only one block of B when ICASX = 3 or 4

NC = row dimension of CC (C transposed)

ND = row dimension of CD (D transposed)

NEQN = starting column of D_{ws}, relative to start of C

NEQP = starting column of zeros after D'_{ww}, relative to start of D

NEQS = starting column of D'_{ww}, relative to start of D

NEQSP = starting column of D'_{ww}, relative to start of C

RKHX = minimum range for using the lumped current approximation for the field of a segment

```
      SUBROUTINE CMNGF (CB,CC,CD,NB,NC,ND,RKHX,IEXKX)                   CG   1
C     CMNGF FILLS INTERACTION MATRICIES B, C, AND D FOR N.G.F. SOLUTION CG   2
      COMPLEX CB,CC,CD,ZARRAY,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC       CG   3
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 CG   4
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( CG   5
     2300),WLAM,IPSYM                                                   CG   6
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF                            CG   7
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP CG   8
     1CON(10),NPCON                                                     CG   9
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ CG  10
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND                            CG  11
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I CG  12
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL                                CG  13
      DIMENSION CB(NB,1), CC(NC,1), CD(ND,1)                            CG  14
      RKH=RKHX                                                          CG  15
      IEXK=IEXKX                                                        CG  16
      M1EQ=2*M1                                                         CG  17
      M2EQ=M1EQ+1                                                       CG  18
      MEQ=2*M                                                           CG  19
      NEQP=ND-NPCON*2                                                   CG  20
      NEQS=NEQP-NSCON                                                   CG  21
      NEQSP=NEQS+NC                                                     CG  22
      NEQN=NC+N-N1                                                      CG  23
      ITX=1                                                             CG  24
      IF (NSCON.GT.0) ITX=2                                             CG  25
      IF (ICASX.EQ.1) GO TO 1                                           CG  26
      REWIND 12                                                         CG  27
      REWIND 14                                                         CG  28
      REWIND 15                                                         CG  29
      IF (ICASX.GT.2) GO TO 5                                           CG  30
1     DO 4 J=1,ND                                                       CG  31
      DO 2 I=1,ND                                                       CG  32
2     CD(I,J)=(0.,0.)                                                   CG  33
      DO 3 I=1,NB                                                       CG  34
      CB(I,J)=(0.,0.)                                                   CG  35
3     CC(I,J)=(0.,0.)                                                   CG  36
4     CONTINUE                                                          CG  37
5     IST=N-N1+1                                                        CG  38
      IT=NPBX                                                           CG  39
      ISV=-NPBX                                                         CG  40
C     LOOP THRU 24 FILLS B.  FOR ICASX=1 OR 2 ALSO FILLS D(WW), D(WS)   CG  41
      DO 24 IBLK=1,NBBX                                                 CG  42
      ISV=ISV+NPBX                                                      CG  43
      IF (IBLK.EQ.NBBX) IT=NLBX                                         CG  44
      IF (ICASX.LT.3) GO TO 7                                           CG  45
      DO 6 J=1,ND                                                       CG  46
      DO 6 I=1,IT                                                       CG  47
6     CB(I,J)=(0.,0.)                                                   CG  48
7     I1=ISV+1                                                          CG  49
      I2=ISV+IT                                                         CG  50
      IN2=I2                                                            CG  51
      IF (IN2.GT.N1) IN2=N1                                             CG  52
      IM1=I1-N1                                                         CG  53
      IM2=I2-N1                                                         CG  54
      IF (IM1.LT.1) IM1=1                                               CG  55
      IMX=1                                                             CG  56
      IF (I1.LE.N1) IMX=N1-I1+2                                         CG  57
      IF (N2.GT.N) GO TO 12                                             CG  58
C     FILL B(WW),B(WS).  FOR ICASX=1,2 FILL D(WW),D(WS)                 CG  59
      DO 11 J=N2,N                                                      CG  60
      CALL TRIO (J)                                                     CG  61
      DO 9 I=1,JSNO                                                     CG  62
      JSS=JCO(I)                                                        CG  63
      IF (JSS.LT.N2) GO TO 8                                            CG  64
```

```
C     SET JCO WHEN SOURCE IS NEW BASIS FUNCTION ON NEW SEGMENT          CG  65
      JCO(I)=JSS-N1                                                     CG  66
      GO TO 9                                                           CG  67
C     SOURCE IS PORTION OF MODIFIED BASIS FUNCTION ON NEW SEGMENT       CG  68
8     JCO(I)=NEQS+ICONX(JSS)                                            CG  69
9     CONTINUE                                                          CG  70
      IF (I1.LE.IN2) CALL CMWW (J,I1,IN2,CB,NB,CB,NB,0)                 CG  71
      IF (IM1.LE.IM2) CALL CMWS (J,IM1,IM2,CB(IMX,1),NB,CB,NB,0)        CG  72
      IF (ICASX.GT.2) GO TO 11                                          CG  73
      CALL CMWW (J,N2,N,CD,ND,CD,ND,1)                                  CG  74
      IF (M2.LE.M) CALL CMWS (J,M2EQ,MEQ,CD(1,IST),ND,CD,ND,1)          CG  75
C     LOADING IN D(WW)                                                  CG  76
      IF (NLOAD.EQ.0) GO TO 11                                          CG  77
      IR=J-N1                                                           CG  78
      EXK=ZARRAY(J)                                                     CG  79
      DO 10 I=1,JSNO                                                    CG  80
      JSS=JCO(I)                                                        CG  81
10    CD(JSS,IR)=CD(JSS,IR)-(AX(I)+CX(I))*EXK                           CG  82
11    CONTINUE                                                          CG  83
12    IF (NSCON.EQ.0) GO TO 20                                          CG  84
C     FILL B(WW)PRIME                                                   CG  85
      DO 19 I=1,NSCON                                                   CG  86
      J=ISCON(I)                                                        CG  87
C     SOURCES ARE NEW OR MODIFIED BASIS FUNCTIONS ON OLD SEGMENTS WHICH CG  88
C     CONNECT TO NEW SEGMENTS                                           CG  89
      CALL TRIO (J)                                                     CG  90
      JSS=0                                                             CG  91
      DO 15 IX=1,JSNO                                                   CG  92
      IR=JCO(IX)                                                        CG  93
      IF (IR.LT.N2) GO TO 13                                            CG  94
      IR=IR-N1                                                          CG  95
      GO TO 14                                                          CG  96
13    IR=ICONX(IR)                                                      CG  97
      IF (IR.EQ.0) GO TO 15                                             CG  98
      IR=NEQS+IR                                                        CG  99
14    JSS=JSS+1                                                         CG 100
      JCO(JSS)=IR                                                       CG 101
      AX(JSS)=AX(IX)                                                    CG 102
      BX(JSS)=BX(IX)                                                    CG 103
      CX(JSS)=CX(IX)                                                    CG 104
15    CONTINUE                                                          CG 105
      JSNO=JSS                                                          CG 106
      IF (I1.LE.IN2) CALL CMWW (J,I1,IN2,CB,NB,CB,NB,0)                 CG 107
      IF (IM1.LE.IM2) CALL CMWS (J,IM1,IM2,CB(IMX,1),NB,CB,NB,0)        CG 108
C     SOURCE IS SINGULAR COMPONENT OF PATCH CURRENT THAT IS PART OF     CG 109
C     MODIFIED BASIS FUNCTION FOR OLD SEGMENT THAT CONNECTS TO A NEW    CG 110
C     SEGMENT ON END OPPOSITE PATCH.                                    CG 111
      IF (I1.LE.IN2) CALL CMSW (J,I,I1,IN2,CB,CB,0,NB,-1)               CG 112
      IF (NLODF.EQ.0) GO TO 17                                          CG 113
      JX=J-ISV                                                          CG 114
      IF (JX.LT.1.OR.JX.GT.IT) GO TO 17                                 CG 115
      EXK=ZARRAY(J)                                                     CG 116
      DO 16 IX=1,JSNO                                                   CG 117
      JSS=JCO(IX)                                                       CG 118
16    CB(JX,JSS)=CB(JX,JSS)-(AX(IX)+CX(IX))*EXK                         CG 119
C     SOURCES ARE PORTIONS OF MODIFIED BASIS FUNCTION J ON OLD SEGMENTS CG 120
C     EXCLUDING OLD SEGMENTS THAT DIRECTLY CONNECT TO NEW SEGMENTS.     CG 121
17    CALL TBF (J,1)                                                    CG 122
      JSX=JSNO                                                          CG 123
      JSNO=1                                                            CG 124
      IR=JCO(1)                                                         CG 125
      JCO(1)=NEQS+I                                                     CG 126
      DO 19 IX=1,JSX                                                    CG 127
      IF (IX.EQ.1) GO TO 18                                             CG 128
```

```
      IR=JCO(IX)
      AX(1)=AX(IX)
      BX(1)=BX(IX)
      CX(1)=CX(IX)
   18 IF (IR.GT.N1) GO TO 19
      IF (ICONX(IR).NE.0) GO TO 19
      IF (I1.LE.IN2) CALL CMWW (IR,I1,IN2,CB,NB,CB,NB,0)
      IF (IM1.LE.IM2) CALL CMWS (IR,IM1,IM2,CB(IMX,1),NB,CB,NB,0)
C     LOADING FOR B(WW)PRIME
      IF (NLODF.EQ.0) GO TO 19
      JX=IR-ISV
      IF (JX.LT.1.OR.JX.GT.IT) GO TO 19
      EXK=ZARRAY(IR)
      JSS=JCO(1)
      CB(JX,JSS)=CB(JX,JSS)-(AX(1)+CX(1))*EXK
   19 CONTINUE
   20 IF (NPCON.EQ.0) GO TO 22
      JSS=NEQP
C     FILL B(SS)PRIME TO SET OLD PATCH BASIS FUNCTIONS TO ZERO FOR
C     PATCHES THAT CONNECT TO NEW SEGMENTS
      DO 21 I=1,NPCON
      IX=IPCON(I)*2+N1-ISV
      IR=IX-1
      JSS=JSS+1
      IF (IR.GT.0.AND.IR.LE.IT) CB(IR,JSS)=(1.,0.)
      JSS=JSS+1
      IF (IX.GT.0.AND.IX.LE.IT) CB(IX,JSS)=(1.,0.)
   21 CONTINUE
   22 IF (M2.GT.M) GO TO 23
C     FILL B(SW) AND B(SS)
      IF (I1.LE.IN2) CALL CMSW (M2,M,I1,IN2,CB(1,IST),CB,N1,NB,0)
      IF (IM1.LE.IM2) CALL CMSS (M2,M,IM1,IM2,CB(IMX,IST),NB,0)
   23 IF (ICASX.EQ.1) GO TO 24
      WRITE (14) ((CB(I,J),I=1,IT),J=1,ND)
   24 CONTINUE
C     FILLING B COMPLETE.  START ON C AND D
      IT=NPBL
      ISV=-NPBL
      DO 43 IBLK=1,NBBL
      ISV=ISV+NPBL
      ISVV=ISV+NC
      IF (IBLK.EQ.NBBL) IT=NLBL
      IF (ICASX.LT.3) GO TO 27
      DO 26 J=1,IT
      DO 25 I=1,NC
   25 CC(I,J)=(0.,0.)
      DO 26 I=1,ND
   26 CD(I,J)=(0.,0.)
   27 I1=ISVV+1
      I2=ISVV+IT
      IN1=I1-M1EQ
      IN2=I2-M1EQ
      IF (IN2.GT.N) IN2=N
      IM1=I1-N
      IM2=I2-N
      IF (IM1.LT.M2EQ) IM1=M2EQ
      IF (IM2.GT.MEQ) IM2=MEQ
      IMX=1
      IF (IN1.LE.IN2) IMX=NEQN-I1+2
      IF (ICASX.LT.3) GO TO 32
      IF (N2.GT.N) GO TO 32
C     SAME AS DO 24 LOOP TO FILL D(WW) FOR ICASX GREATER THAN 2
      DO 31 J=N2,N
      CALL TRIO (J)
```

```
      DO 29 I=1,JSNO                                                    CG 193
      JSS=JCO(I)                                                        CG 194
      IF (JSS.LT.N2) GO TO 28                                           CG 195
      JCO(I)=JSS-N1                                                     CG 196
      GO TO 29                                                          CG 197
   28 JCO(I)=NEQS+ICONX(JSS)                                            CG 198
   29 CONTINUE                                                          CG 199
      IF (IN1.LE.IN2) CALL CMWW (J,IN1,IN2,CD,ND,CD,ND,1)               CG 200
      IF (IM1.LE.IM2) CALL CMWS (J,IM1,IM2,CD(1,IMX),ND,CD,ND,1)        CG 201
      IF (NLOAD.EQ.0) GO TO 31                                          CG 202
      IR=J-N1-ISV                                                       CG 203
      IF (IR.LT.1.OR.IR.GT.IT) GO TO 31                                 CG 204
      EXK=ZARRAY(J)                                                     CG 205
      DO 30 I=1,JSNO                                                    CG 206
      JSS=JCO(I)                                                        CG 207
   30 CD(JSS,IR)=CD(JSS,IR)-(AX(I)+CX(I))*EXK                           CG 208
   31 CONTINUE                                                          CG 209
   32 IF (M2.GT.M) GO TO 33                                             CG 210
C     FILL D(SW) AND D(SS)                                              CG 211
      IF (IN1.LE.IN2) CALL CMSW (M2,M,IN1,IN2,CD(IST,1),CD,N1,ND,1)     CG 212
      IF (IM1.LE.IM2) CALL CMSS (M2,M,IM1,IM2,CD(IST,IMX),ND,1)         CG 213
   33 IF (N1.LT.1) GO TO 39                                             CG 214
C     FILL C(WW),C(WS), D(WW)PRIME, AND D(WS)PRIME.                     CG 215
      DO 37 J=1,N1                                                      CG 216
      CALL TRIO (J)                                                     CG 217
      IF (NSCON.EQ.0) GO TO 36                                          CG 218
      DO 35 IX=1,JSNO                                                   CG 219
      JSS=JCO(IX)                                                       CG 220
      IF (JSS.LT.N2) GO TO 34                                           CG 221
      JCO(IX)=JSS+M1EQ                                                  CG 222
      GO TO 35                                                          CG 223
   34 IR=ICONX(JSS)                                                     CG 224
      IF (IR.NE.0) JCO(IX)=NEQSP+IR                                     CG 225
   35 CONTINUE                                                          CG 226
   36 IF (IN1.LE.IN2) CALL CMWW (J,IN1,IN2,CC,NC,CD,ND,ITX)             CG 227
      IF (IM1.LE.IM2) CALL CMWS (J,IM1,IM2,CC(1,IMX),NC,CD(1,IMX),ND,ITX CG 228
     1)                                                                 CG 229
   37 CONTINUE                                                          CG 230
      IF (NSCON.EQ.0) GO TO 39                                          CG 231
C     FILL C(WW)PRIME                                                   CG 232
      DO 38 IX=1,NSCON                                                  CG 233
      IR=ISCON(IX)                                                      CG 234
      JSS=NEQS+IX-ISV                                                   CG 235
      IF (JSS.GT.0.AND.JSS.LE.IT) CC(IR,JSS)=(1.,0.)                    CG 236
   38 CONTINUE                                                          CG 237
   39 IF (NPCON.EQ.0) GO TO 41                                          CG 238
      JSS=NEQP-ISV                                                      CG 239
C     FILL C(SS)PRIME                                                   CG 240
      DO 40 I=1,NPCON                                                   CG 241
      IX=IPCON(I)*2+N1                                                  CG 242
      IR=IX-1                                                           CG 243
      JSS=JSS+1                                                         CG 244
      IF (JSS.GT.0.AND.JSS.LE.IT) CC(IR,JSS)=(1.,0.)                    CG 245
      JSS=JSS+1                                                         CG 246
      IF (JSS.GT.0.AND.JSS.LE.IT) CC(IX,JSS)=(1.,0.)                    CG 247
   40 CONTINUE                                                          CG 248
   41 IF (M1.LT.1) GO TO 42                                             CG 249
C     FILL C(SW) AND C(SS)                                              CG 250
      IF (IN1.LE.IN2) CALL CMSW (1,M1,IN1,IN2,CC(N2,1),CC,0,NC,1)       CG 251
      IF (IM1.LE.IM2) CALL CMSS (1,M1,IM1,IM2,CC(N2,IMX),NC,1)          CG 252
   42 CONTINUE                                                          CG 253
      IF (ICASX.EQ.1) GO TO 43                                          CG 254
      WRITE (12) ((CD(J,I),J=1,ND),I=1,IT)                              CG 255
      WRITE (15) ((CC(J,I),J=1,NC),I=1,IT)                              CG 256
```

```
 43     CONTINUE                                                      CG 257
        IF(ICASX.EQ.1)RETURN                                          CG 258
        REWIND 12                                                     CG 259
        REWIND 14                                                     CG 260
        REWIND 15                                                     CG 261
        RETURN                                                        CG 262
        END                                                           CG 263-
```

CMSET

PURPOSE

To control the filling of the interaction matrix.

METHOD

The linear equations resulting from the moment method solution of equations 13, 14 and the negative of equation 15 in Part I are written as

$$\sum_{j=1}^{N} a_j A_{ij} + \sum_{j=1}^{2M} b_j B_{ij} = E_i, \qquad i = 1, \ldots N$$

$$\sum_{j=1}^{N} c_j C_{kj} + \sum_{j=1}^{2M} d_j D_{kj} = H_k, \qquad k = 1, \ldots 2M$$

where N = number of segments
M = number of patches

$A_{ij} = \hat{s}_i \cdot (\bar{E}$ at $\bar{r}_i$ due to segment basis function j)

$B_{ij} = \hat{s}_i \cdot (\bar{E}$ at $\bar{r}_i$ due to current on patch $[(j + 1)/2]$ in direction $\hat{u}_j$)

$C_{kj} = -\hat{v}_k \cdot (\bar{H}$ at $\bar{p}_{[(k+1)/2]}$ due to segment basis function j) $\cdot S_{[(k+1)/2]}$

$D_{kj} = -\hat{v}_k \cdot (\bar{H}$ at $\bar{p}_{[(k+1)/2]}$ due to current on patch $[(j+1)/2]$ in direction $\hat{u}_j$) $S_{[(k+1)/2]} + \frac{1}{2}\sigma_{kj}$

$E_i = -\hat{s}_i \cdot$ (incident electric field at $\bar{r}_i$)

$H_k = \hat{v}_k \cdot$ (incident magnetic field at $\bar{p}_{[(k+1)/2]}$) $S_{[(k+1)/2]}$

$\bar{r}_i$ = position of the center of segment i

$\bar{p}_i$ = position of the center of patch i

$\hat{s}_i$ = unit vector in the direction of segment i

$$\hat{u}_i = \left\{ \begin{array}{ll} \hat{t}_1 & \text{if i is odd} \\ \hat{t}_2 & \text{if i is even} \end{array} \right\} \text{ for patch } [(i+1)/2]$$

$$\hat{v}_i = \left\{ \begin{array}{ll} \hat{t}_2 & \text{if i is odd} \\ \hat{t}_1 & \text{if i is even} \end{array} \right\} \text{ for patch } [(i+1)/2]$$

$S_i = 1$ if $\hat{t}_1 \times \hat{t}_2 = \hat{n}$ on patch i

$\quad -1$ if $\hat{t}_1 \times \hat{t}_2 = -\hat{n}$ on patch i

$\sigma_{kj} = -1$ if $k = j =$ odd

$\quad +1$ if $k = j =$ even

$\quad 0$ if $k \neq j$

The basis function amplitudes a_j, b_j, c_j and d_j are determined later by solving the matrix equation of order N + 2M.

The matrix elements are computed by calling subroutines CMWW, CMSW, CMWS, and CMSS for the elements of A, B, C and D respectively. For A and C the components of all basis functions that extend across segment J are computed by calling TRIO at CM 52. CMWW and CMWS are then called to compute the components of A or C due to these basis function components on segment J.

If segment j, with length Δ_j, is loaded with impedance Z_j the elements of A are modified as $A_{jk} = A_{jk} - \frac{Z_j}{\Delta j}$ X (value of basis function k at the center of segment j) for k = the numbers of all basis functions that extend onto segment j. The summation over values of k (k = JSS) for loading on segment J occurs at CM 68.

The submatrices are stored in the array CM in transposed form. All references to rows and columns, here, apply to the nontransposed matrices. Thus "row" in this discussion refers to the second index of CM in the code.

For a structure without symmetry the submatrices are stored in the order

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

If the complete matrix is too large for the array CM then blocks of rows are filled and written on file 11. A block may then contain rows from A and B, rows from C and D or a combination. The row of CM at which C and D start is computed as IST.

For a structure having p symmetric sections the submatrices are stored in the form

$$\begin{bmatrix} A_1 & B_1 & A_2 & B_2 & & A_p & B_p \\ & & & & \cdots & & \\ C_1 & D_1 & C_2 & D_2 & & C_p & D_p \end{bmatrix}$$

where $\begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$

represents A_i in the first row of submatrices in equation 108 of Part I. Each call to CMWW and CMWS may fill elements of A_i or C_i for any value of i. The column indices in array JCO are adjusted at CM 55 to allow for the columns occupied by the B_i and D_i matrices. B_i and D_i are filled for each value of i in the loop from CM 75 to CM 81. The Fourier transform of the submatrices, or the transform for planar symmetry (equation 116 of Part I) is computed from CM 85 to CM 100.

SYMBOL DICTIONARY

CM = array for the matrix

I1 = number of first equation in a block (patch equation +N for patches)

I2 = number of the last equation in a block

IEXKX = 1 to use extended thin wire kernel on wires, 0 otherwise

IM1 = number of first patch equation in a block

CMSET

IM2	=	number of last patch equation in a block
IN2	=	number of the last segment equation in a block
IOUT	=	number of real numbers in a block for output
IPR	=	row in CM (second index) for segment J
IST	=	row in CM of the first patch equation
ISV	=	I1 - 1
IT	=	number of rows in a block
IXBLK1	=	block number
JM1	=	number of first patch in a symmetric section
JM2	=	number of the last patch in a symmetric section
JST	=	column in CM of the first patch equation for a symmetric block
MP2	=	number of patch equations
NEQ	=	total number of equations
NOP	=	number of symmetric sections
NPEQ	=	number of equations in a symmetric section
NROW	=	row dimensions of the transposed CM array
RKHX	=	minimum interaction distance at which the infinitesimal dipole approximation is used for the field of a segment
ZAJ	=	Z_j/Δ_j

```
      SUBROUTINE CMSET (NROW,CM,RKHX,IEXKX)
C
C     CMSET SETS UP THE COMPLEX STRUCTURE MATRIX IN THE ARRAY CM
C
      COMPLEX CM,ZARRAY,ZAJ,ETK,ETS,ETC,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,
     1EZC,SSX,D,DETER
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      COMMON /SMAT/ SSX(16,16)
      COMMON /SCRATM/ D(600)
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP
     1CON(10),NPCON
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      DIMENSION CM(NROW,1)
      MP2=2*MP
      NPEQ=NP+MP2
      NEQ=N+2*M
      NOP=NEQ/NPEQ
      IF (ICASE.GT.2) REWIND 11
      RKH=RKHX
      IEXK=IEXKX
      IOUT=2*NPBLK*NROW
      IT=NPBLK
C
C     CYCLE OVER MATRIX BLOCKS
C
      DO 13 IXBLK1=1,NBLOKS
      ISV=(IXBLK1-1)*NPBLK
      IF (IXBLK1.EQ.NBLOKS) IT=NLAST
      DO 1 I=1,NROW
      DO 1 J=1,IT
1     CM(I,J)=(0.,0.)
      I1=ISV+1
      I2=ISV+IT
      IN2=I2
      IF (IN2.GT.NP) IN2=NP
      IM1=I1-NP
      IM2=I2-NP
      IF (IM1.LT.1) IM1=1
      IST=1
      IF (I1.LE.NP) IST=NP-I1+2
      IF (N.EQ.0) GO TO 5
C
C     WIRE SOURCE LOOP
C
      DO 4 J=1,N
      CALL TRIO (J)
      DO 2 I=1,JSNO
      IJ=JCO(I)
2     JCO(I)=((IJ-1)/NP)*MP2+IJ
      IF (I1.LE.IN2) CALL CMWW (J,I1,IN2,CM,NROW,CM,NROW,1)
      IF (IM1.LE.IM2) CALL CMWS (J,IM1,IM2,CM(1,IST),NROW,CM,NROW,1)
      IF (NLOAD.EQ.0) GO TO 4
C
C     MATRIX ELEMENTS MODIFIED BY LOADING
C
      IF (J.GT.NP) GO TO 4
      IPR=J-ISV
      IF (IPR.LT.1.OR.IPR.GT.IT) GO TO 4
```

```
      ZAJ=ZARRAY(J)                                                     CM  65
      DO 3 I=1,JSNO                                                     CM  66
      JSS=JCO(I)                                                        CM  67
    3 CM(JSS,IPR)=CM(JSS,IPR)-(AX(I)+CX(I))*ZAJ                         CM  68
    4 CONTINUE                                                          CM  69
    5 IF (M.EQ.0) GO TO 7                                               CM  70
C     MATRIX ELEMENTS FOR PATCH CURRENT SOURCES                         CM  71
      JM1=1-MP                                                          CM  72
      JM2=0                                                             CM  73
      JST=1-MP2                                                         CM  74
      DO 6 I=1,NOP                                                      CM  75
      JM1=JM1+MP                                                        CM  76
      JM2=JM2+MP                                                        CM  77
      JST=JST+NPEQ                                                      CM  78
      IF (I1.LE.IN2) CALL CMSW (JM1,JM2,I1,IN2,CM(JST,1),CM,0,NROW,1)   CM  79
      IF (IM1.LE.IM2) CALL CMSS (JM1,JM2,IM1,IM2,CM(JST,IST),NROW,1)    CM  80
    6 CONTINUE                                                          CM  81
    7 IF (ICASE.EQ.1) GO TO 13                                          CM  82
      IF (ICASE.EQ.3) GO TO 12                                          CM  83
C     COMBINE ELEMENTS FOR SYMMETRY MODES                               CM  84
      DO 11 I=1,IT                                                      CM  85
      DO 11 J=1,NPEQ                                                    CM  86
      DO 8 K=1,NOP                                                      CM  87
      KA=J+(K-1)*NPEQ                                                   CM  88
    8 D(K)=CM(KA,I)                                                     CM  89
      DETER=D(1)                                                        CM  90
      DO 9 KK=2,NOP                                                     CM  91
    9 DETER=DETER+D(KK)                                                 CM  92
      CM(J,I)=DETER                                                     CM  93
      DO 11 K=2,NOP                                                     CM  94
      KA=J+(K-1)*NPEQ                                                   CM  95
      DETER=D(1)                                                        CM  96
      DO 10 KK=2,NOP                                                    CM  97
   10 DETER=DETER+D(KK)*SSX(K,KK)                                       CM  98
      CM(KA,I)=DETER                                                    CM  99
   11 CONTINUE                                                          CM 100
      IF (ICASE.LT.3) GO TO 13                                          CM 101
C     WRITE BLOCK FOR OUT-OF-CORE CASES.                                CM 102
   12 CALL BLCKOT (CM,11,1,IOUT,1,31)                                   CM 103
   13 CONTINUE                                                          CM 104
      IF (ICASE.GT.2) REWIND 11                                         CM 105
      RETURN                                                            CM 106
      END                                                               CM 107-
```

CMSS

PURPOSE

To compute and store matrix elements representing the H field at patch centers due to the current on patches.

METHOD

CMSS computes the matrix elements D_{kj} defined in the description of subroutine CMSET. Subroutine HINTG is called to compute the magnetic field at the center of patch I due to current on patch J. H due to the current $\hat{t}_1$ on patch J is stored in EXK, EYK and EZK, while H due to current $\hat{t}_2$ is stored in EXS, EYS and EZS. The term 0.5 σ_{kj} in D_{kj} is added at CM 61 and CM 62 for odd and even equations. The matrix elements are stored in array CM from SS63 to SS78 in either normal or transposed order. Elements for both the even and odd equations are stored if both equations are within the block.

SYMBOL DICTIONARY

CM	= array for matrix storage
G11	= D_{kj} for k odd, j odd
G12	= D_{kj} for k odd, j even
G21	= D_{kj} for k even, j odd
G22	= D_{kj} for k even, j even
I1	= patch number for first equation
I2	= patch number for last equation
ICOMP	= equation number for the odd numbered equation for observation patch I
II1	= location of the odd numbered equation in CM
II2	= location of the even numbered equation in CM
IL	= array location for coordinates of patch I
IM1	= patch equation number for first equation in block
IM2	= patch equation number for last equation in block
ITRP	= 0 or 1 to select normal or transposed filling of CM
J1	= number of first source patch
J2	= number of last source patch

JJ1	= column in non-transposed matrix, of the first equation for patch J
JJ2	= column of second equation for patch J
JL	= array location for coordinates of patch J
NROW	= row dimension of CM
T1XI, T1YI, T1ZI T2XI, T2YI, T2ZI T1XJ, T1YJ, T1ZJ T2XJ, T2YJ, T2ZJ	= x, y and z components of $\hat{t}_1$ or $\hat{t}_2$ for patch I or J
XI, YI, ZI	= coordinates of center of patch I

```
      SUBROUTINE CMSS (J1,J2,IM1,IM2,CM,NROW,ITRP)
C     CMSS COMPUTES MATRIX ELEMENTS FOR SURFACE-SURFACE INTERACTIONS.
      COMPLEX G11,G12,G21,G22,CM,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      DIMENSION CM(NROW,1)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y
     1J,IND1), (T2ZJ,IND2)
      LDP=LD+1
      I1=(IM1+1)/2
      I2=(IM2+1)/2
      ICOMP=I1*2-3
      II1=-1
      IF (ICOMP+2.LT.IM1) II1=-2
C     LOOP OVER OBSERVATION PATCHES
      DO 5 I=I1,I2
      IL=LDP-I
      ICOMP=ICOMP+2
      II1=II1+2
      II2=II1+1
      T1XI=T1X(IL)*SALP(IL)
      T1YI=T1Y(IL)*SALP(IL)
      T1ZI=T1Z(IL)*SALP(IL)
      T2XI=T2X(IL)*SALP(IL)
      T2YI=T2Y(IL)*SALP(IL)
      T2ZI=T2Z(IL)*SALP(IL)
      XI=X(IL)
      YI=Y(IL)
      ZI=Z(IL)
      JJ1=-1
C     LOOP OVER SOURCE PATCHES
      DO 5 J=J1,J2
      JL=LDP-J
      JJ1=JJ1+2
      JJ2=JJ1+1
      S=BI(JL)
      XJ=X(JL)
      YJ=Y(JL)
      ZJ=Z(JL)
      T1XJ=T1X(JL)
      T1YJ=T1Y(JL)
      T1ZJ=T1Z(JL)
      T2XJ=T2X(JL)
      T2YJ=T2Y(JL)
      T2ZJ=T2Z(JL)
      CALL HINTG (XI,YI,ZI)
      G11=-(T2XI*EXK+T2YI*EYK+T2ZI*EZK)
      G12=-(T2XI*EXS+T2YI*EYS+T2ZI*EZS)
      G21=-(T1XI*EXK+T1YI*EYK+T1ZI*EZK)
      G22=-(T1XI*EXS+T1YI*EYS+T1ZI*EZS)
      IF (I.NE.J) GO TO 1
      G11=G11-.5
      G22=G22+.5
    1 IF (ITRP.NE.0) GO TO 3
C     NORMAL FILL
      IF (ICOMP.LT.IM1) GO TO 2
      CM(II1,JJ1)=G11
```

```
      CM(II1,JJ2)=G12
    2 IF (ICOMP.GE.IM2) GO TO 5
      CM(II2,JJ1)=G21
      CM(II2,JJ2)=G22
      GO TO 5
C     TRANSPOSED FILL
    3 IF (ICOMP.LT.IM1) GO TO 4
      CM(JJ1,II1)=G11
      CM(JJ2,II1)=G12
    4 IF (ICOMP.GE.IM2) GO TO 5
      CM(JJ1,II2)=G21
      CM(JJ2,II2)=G22
    5 CONTINUE
      RETURN
      END
```

CMSW

PURPOSE

To compute and store matrix elements representing the electric field at segment centers due to the current on patches.

METHOD

SW30 - SW35	Coordinates of observation segment are stored.
SW36 - SW42	If either end of the observation segment connects to a surface IPCH is set to the number of the first of the four patches at the connection point.
SW48 - SW57	Coordinates of the source patch are stored in COMMON/DATAJ/.
SW61 - SW86	IF IPCH = J then patch J is the first patch at the point where segment I connects to the surface. Subroutine PCINT is called to integrate the current over the four patches at the connection point. The current on the patches includes the eight basis functions of the four patches and a portion of the basis function from the segment. Hence contributions to nine matrix elements are generated and stored in array EMEL. The field due to the segment basis function extending onto the patches is stored in array CW at SW76 or SW78. The fields due to the first patch basis function, EMEL(1) and EMEL(5), are then stored in array CM at SW80 and SW81 or at SW83 and SW84. ICGO is then incremented. For the next three times through the loop over J the call to PCINT is skipped at SW63 and the remaining values in EMEL are stored.
SW88 - SW96	If segment I and patch J are not connected, subroutine UNERE is called to compute the electric field due to the current on the patch with the current treated as Hertzian dipoles in the directions $\hat{t}_1$ and $\hat{t}_2$. The matrix elements are stored in CM.

SW102 - SW138 This is a special section of code to compute the electric field due to the component of a segment basis function that extends onto connected patches. It is used at line CG112 of subroutine CMNGF for the case where the connected segment and patches are in the NGF file and a new segment is connected to the outer end of the NGF segment modifying its basis function. Subroutine PCINT is called to evaluate the nine matrix elements. Only EMEL(9) is used since the patch basis functions have not been modified.

SYMBOL DICTIONARY

CABI	= x component of $\hat{i}$ in direction of segment I
CM	= array for E due to patch basis functions
CW	= array for E due to segment basis function extending onto surface at connection point
EMEL	= array of matrix elements from integrating over surface
FSIGN	= ± 1 depending on which end of segment connects to surface
I1	= number of first observation segment
I2	= number of last observation segment
ICGØ	= index for matrix elements at connection point
IL	= index for segment basis function in CW
IP	= 1 for direct field, 2 for image in ground
IPCH	= number of first patch connecting to a segment
ITRP	= 0 for normal matrix fill 1 for transposed fill -1 for special NGF case
J	= source patch
J1	= first source patch
J2	= last source patch
JL	= index for source patch in CM
JS	= index for patch coordinates
K	= index in CM or CW for observation segment
NCW	= index offset for CW

NEQS	= number of equations excluding NGF
NROW	= row dimensions of CM and CW
PI	= pi
PX	= sin $k(s - s_0)$ for s at the end of the segment
PY	= cos $k(s - s_0)$ connected to the surface
SABI	= y component of $\hat{i}$ in direction of segment I
SALPI	= z component of $\hat{i}$ in direction of segment I
XI, YI, ZI	= center of observation segment

CMSW

```
      SUBROUTINE CMSW (J1,J2,I1,I2,CM,CW,NCW,NROW,ITRP)                 SW   1
C     COMPUTES MATRIX ELEMENTS FOR E ALONG WIRES DUE TO PATCH CURRENT   SW   2
      COMPLEX CM,ZRATI,ZRATI2,T1,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC,EME SW   3
     1L,CW,FRATI                                                        SW   4
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 SW   5
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( SW   6
     2300),WLAM,IPSYM                                                   SW   7
      COMMON /ANGL/ SALP(300)                                           SW   8
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR, SW   9
     1IPERF,T1,T2                                                       SW  10
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ SW  11
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND                            SW  12
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP SW  13
     1CON(10),NPCON                                                     SW  14
      DIMENSION CAB(1), SAB(1), CM(NROW,1), CW(NROW,1)                  SW  15
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), EMEL(9) SW  16
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON SW  17
     12), (T2Z,ITAG), (CAB,ALP), (SAB,BET)                              SW  18
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y SW  19
     1J,IND1), (T2ZJ,IND2)                                              SW  20
      DATA PI/3.141592654/                                              SW  21
      LDP=LD+1                                                          SW  22
      NEQS=N-N1+2*(M-M1)                                                SW  23
      IF (ITRP.LT.0) GO TO 13                                           SW  24
      K=0                                                               SW  25
      ICGO=1                                                            SW  26
C     OBSERVATION LOOP                                                  SW  27
      DO 12 I=I1,I2                                                     SW  28
      K=K+1                                                             SW  29
      XI=X(I)                                                           SW  30
      YI=Y(I)                                                           SW  31
      ZI=Z(I)                                                           SW  32
      CABI=CAB(I)                                                       SW  33
      SABI=SAB(I)                                                       SW  34
      SALPI=SALP(I)                                                     SW  35
      IPCH=0                                                            SW  36
      IF (ICON1(I).LT.10000) GO TO 1                                    SW  37
      IPCH=ICON1(I)-10000                                               SW  38
      FSIGN=-1.                                                         SW  39
1     IF (ICON2(I).LT.10000) GO TO 2                                    SW  40
      IPCH=ICON2(I)-10000                                               SW  41
      FSIGN=1.                                                          SW  42
2     JL=0                                                              SW  43
C     SOURCE LOOP                                                       SW  44
      DO 12 J=J1,J2                                                     SW  45
      JS=LDP-J                                                          SW  46
      JL=JL+2                                                           SW  47
      T1XJ=T1X(JS)                                                      SW  48
      T1YJ=T1Y(JS)                                                      SW  49
      T1ZJ=T1Z(JS)                                                      SW  50
      T2XJ=T2X(JS)                                                      SW  51
      T2YJ=T2Y(JS)                                                      SW  52
      T2ZJ=T2Z(JS)                                                      SW  53
      XJ=X(JS)                                                          SW  54
      YJ=Y(JS)                                                          SW  55
      ZJ=Z(JS)                                                          SW  56
      S=BI(JS)                                                          SW  57
C     GROUND LOOP                                                       SW  58
      DO 12 IP=1,KSYMP                                                  SW  59
      IPGND=IP                                                          SW  60
      IF (IPCH.NE.J.AND.ICGO.EQ.1) GO TO 9                              SW  61
      IF (IP.EQ.2) GO TO 9                                              SW  62
      IF (ICGO.GT.1) GO TO 6                                            SW  63
      CALL PCINT (XI,YI,ZI,CABI,SABI,SALPI,EMEL)                        SW  64
```

```
      PY=PI*SI(I)*FSIGN
      PX=SIN(PY)
      PY=COS(PY)
      EXC=EMEL(9)*FSIGN
      CALL TRIO (I)
      IF (I.GT.N1) GO TO 3
      IL=NEQS+ICONX(I)
      GO TO 4
3     IL=I-NCW
      IF (I.LE.NP) IL=((IL-1)/NP)*2*MP+IL
4     IF (ITRP.NE.0) GO TO 5
      CW(K,IL)=CW(K,IL)+EXC*(AX(JSNO)+BX(JSNO)*PX+CX(JSNO)*PY)
      GO TO 6
5     CW(IL,K)=CW(IL,K)+EXC*(AX(JSNO)+BX(JSNO)*PX+CX(JSNO)*PY)
6     IF (ITRP.NE.0) GO TO 7
      CM(K,JL-1)=EMEL(ICGO)
      CM(K,JL)=EMEL(ICGO+4)
      GO TO 8
7     CM(JL-1,K)=EMEL(ICGO)
      CM(JL,K)=EMEL(ICGO+4)
8     ICGO=ICGO+1
      IF (ICGO.EQ.5) ICGO=1
      GO TO 11
9     CALL UNERE (XI,YI,ZI)
      IF (ITRP.NE.0) GO TO 10
C     NORMAL FILL
      CM(K,JL-1)=CM(K,JL-1)+EXK*CABI+EYK*SABI+EZK*SALPI
      CM(K,JL)=CM(K,JL)+EXS*CABI+EYS*SABI+EZS*SALPI
      GO TO 11
C     TRANSPOSED FILL
10    CM(JL-1,K)=CM(JL-1,K)+EXK*CABI+EYK*SABI+EZK*SALPI
      CM(JL,K)=CM(JL,K)+EXS*CABI+EYS*SABI+EZS*SALPI
11    CONTINUE
12    CONTINUE
      RETURN
C     FOR OLD SEG. CONNECTING TO OLD PATCH ON ONE END AND NEW SEG. ON
C     OTHER END INTEGRATE SINGULAR COMPONENT (9) OF SURFACE CURRENT ONLY
13    IF (J1.LT.I1.OR.J1.GT.I2) GO TO 16
      IPCH=ICON1(J1)
      IF (IPCH.LT.10000) GO TO 14
      IPCH=IPCH-10000
      FSIGN=-1.
      GO TO 15
14    IPCH=ICON2(J1)
      IF (IPCH.LT.10000) GO TO 16
      IPCH=IPCH-10000
      FSIGN=1.
15    IF (IPCH.GT.M1) GO TO 16
      JS=LDP-IPCH
      IPGND=1
      T1XJ=T1X(JS)
      T1YJ=T1Y(JS)
      T1ZJ=T1Z(JS)
      T2XJ=T2X(JS)
      T2YJ=T2Y(JS)
      T2ZJ=T2Z(JS)
      XJ=X(JS)
      YJ=Y(JS)
      ZJ=Z(JS)
      S=BI(JS)
      XI=X(J1)
      YI=Y(J1)
      ZI=Z(J1)
      CABI=CAB(J1)
```

```
      SABI=SAB(J1)                                                      SW 129
      SALPI=SALP(J1)                                                    SW 130
      CALL PCINT (XI,YI,ZI,CABI,SABI,SALPI,EMEL)                        SW 131
      PY=PI*SI(J1)*FSIGN                                                SW 132
      PX=SIN(PY)                                                        SW 133
      PY=COS(PY)                                                        SW 134
      EXC=EMEL(9)*FSIGN                                                 SW 135
      IL=JCO(JSNO)                                                      SW 136
      K=J1-I1+1                                                         SW 137
      CW(K,IL)=CW(K,IL)+EXC*(AX(JSNO)+BX(JSNO)*PX+CX(JSNO)*PY)          SW 138
16    RETURN                                                            SW 139
      END                                                               SW 140-
```

CMWS

PURPOSE

To compute and store matrix elements representing the magnetic field at patch centers due to the current on wire segments.

METHOD

Matrix elements are computed for patch equations numbered I1 through I2 with the source segment J. For odd numbered equations the matrix element represents the first term on the right side of equation 14 of Part I. For even numbered equations it is the negative of the first term on the right side of equation 15. For equation I1 and for all odd numbered equations subroutine HSFLD is called to compute the H field at the center of the patch due to constant, sin $k(s - s_0)$ and cos $k(s - s_0)$ currents on segment J. The required component of the field, $-\hat{t}_2 \cdot \bar{H}$ or $-\hat{t}_1 \cdot \bar{H}$ for odd or even equations respectively, is computed from WS49 to WS51. Multiplication by SALP(JS) reverses the sign when $(\hat{t}_1, \hat{t}_2, \hat{n})$ has a left-hand orientation on a patch formed by reflection. The field component for each basis function component on segment J is computed and stored for WS56 through WS75. Storage of the matrix elements is similar to that in subroutine CMWW.

SYMBOL DICTIONARY

CM	= array for matrix elements
CW	= array for matrix elements (NGF only)
ETK, ETS, ETC	= $-\hat{t}_2 \cdot \bar{H}$ or $-\hat{t}_1 \cdot \bar{H}$ due to current of constant, sin $k(s - s_0)$, or cos $k(s - s_0)$ respectively
I	= equation number
I1	= number of first equation
I2	= number of second equation
IK	= 0 if I is even, 1 if I is odd
IPATCH	= patch number for equation I
IPR	= relative matrix location for equation I. Position in complete matrix depends on the address of CM in the call to CMWS

CMWS

ITRP	= 0 for non-transposed fill	
	1 for transposed fill	
	2 for transposed fill for NGF	
J	= source segment number	
JS	= location in COMMON/DATA/ of parameters for patch J	
JX	= matrix index for a particular basis function	
LDP	= LD + 1	
NR	= row dimension of CM	
NW	= row dimension of CW	
TX, TY, TZ	= x, y, and z components of $\hat{t}_1$ or $\hat{t}_2$	
XI, YI, ZI	= x, y and z coordinates of the center of the patch at which the field is computed	

```
      SUBROUTINE CMWS (J,I1,I2,CM,NR,CW,NW,ITRP)
C
C     CMWS COMPUTES MATRIX ELEMENTS FOR WIRE-SURFACE INTERACTIONS
C
      COMPLEX CM,CW,ETK,ETS,ETC,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP
     1CON(10),NPCON
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      DIMENSION CM(NR,1), CW(NW,1), CAB(1), SAB(1)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)
      EQUIVALENCE (CAB,ALP), (SAB,BET), (T1X,SI), (T1Y,ALP), (T1Z,BET)
      EQUIVALENCE (T2X,ICON1), (T2Y,ICON2), (T2Z,ITAG)
      LDP=LD+1
      S=SI(J)
      B=BI(J)
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      CABJ=CAB(J)
      SABJ=SAB(J)
      SALPJ=SALP(J)
C
C     OBSERVATION LOOP
C
      IPR=0
      DO 9 I=I1,I2
      IPR=IPR+1
      IPATCH=(I+1)/2
      IK=I-(I/2)*2
      IF (IK.EQ.0.AND.IPR.NE.1) GO TO 1
      JS=LDP-IPATCH
      XI=X(JS)
      YI=Y(JS)
      ZI=Z(JS)
      CALL HSFLD (XI,YI,ZI,0.)
      IF (IK.EQ.0) GO TO 1
      TX=T2X(JS)
      TY=T2Y(JS)
      TZ=T2Z(JS)
      GO TO 2
1     TX=T1X(JS)
      TY=T1Y(JS)
      TZ=T1Z(JS)
2     ETK=-(EXK*TX+EYK*TY+EZK*TZ)*SALP(JS)
      ETS=-(EXS*TX+EYS*TY+EZS*TZ)*SALP(JS)
      ETC=-(EXC*TX+EYC*TY+EZC*TZ)*SALP(JS)
C
C     FILL MATRIX ELEMENTS.  ELEMENT LOCATIONS DETERMINED BY CONNECTION
C     DATA.
C
      IF (ITRP.NE.0) GO TO 4
C     NORMAL FILL
      DO 3 IJ=1,JSNO
      JX=JCO(IJ)
3     CM(IPR,JX)=CM(IPR,JX)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
      GO TO 9
4     IF (ITRP.EQ.2) GO TO 6
C     TRANSPOSED FILL
      DO 5 IJ=1,JSNO
```

```
      JX=JCO(IJ)
    5 CM(JX,IPR)=CM(JX,IPR)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
      GO TO 9
C     TRANSPOSED FILL - C(WS) AND D(WS)PRIME (=CW)
    6 DO 8 IJ=1,JSNO
      JX=JCO(IJ)
      IF (JX.GT.NR) GO TO 7
      CM(JX,IPR)=CM(JX,IPR)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
      GO TO 8
    7 JX=JX-NR
      CW(JX,IPR)=CW(JX,IPR)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
    8 CONTINUE
    9 CONTINUE
      RETURN
      END
```

CMWW

PURPOSE

To call subroutines to compute the electric field at segment centers due to current on other segments and to store matrix elements in array locations.

METHOD

WW17 - WW24	Parameters of source segment (J) are stored in COMMON/DATAJ/.
WW27 - WW43	First end of segment J is tested to determine whether the extended thin wire approximation can be used. It cannot be used at a junction of more than two wires (WW30), at a bend (WW37), at a change in radius (WW38), or at the base of a non-vertical segment connected to the ground (WW33).
WW44 - WW60	Second end of segment J is tested.
WW66	Loop over observation segments ranges from I1 to I2. The index IPR starts at 1 so the matrix element for I1 is stored in the first row or column of the array CM. The location in the complete matrix is determined by the address given for CM when CMWW is called.
WW76	EFLD computes the electric fields at (XI, YI, ZI) due to segment J and stores them in COMMON/DATAJ/.
WW77 - WW79	Electric field tangent to segment I is computed.
WW84 - WW103	Matrix elements are formed by combining the field components.
WW86 - WW88	Matrix elements are stored in non-transposed order.
WW92 - WW94	Matrix elements are stored in transposed order.
WW97 - WW104	When the source segment is from a NGF file the matrix elements will normally be stored in submatrix C of the NGF matrix structure. When the segment connects to a new segment, however, contributions to submatrix D result. The C and D contributions are stored in CM and CW, respectively, in transposed order.

SYMBOL DICTIONARY

AI	= radius of observation segment
CABI	= x component of unit vector in direction of segment
CM	= array for matrix elements
CW	= array for matrix elements (NGF only)
ETK, ETS, ETC	= E field tangent to segment I due to current of constant, sin $k(s - s_0)$ and cos $k(s - s_0)$ distribution, respectively, on segment J.
I1	= first observation segment
I2	= final observation segment
IJ	= 0 for special treatment when I = J
IPR	= relative matrix location for observation point
ITRP	= 0 for non-transposed fill 1 for transposed fill 2 for transposed fill for NGF
J	= source segment number
JX	= matrix index for a particular basis function
NR	= row dimension of CM
NW	= row dimension of CW
SABI	= y component of unit vector in direction of segment
SALPI	= z component of unit vector in direction of segment
XI, YI, ZI	= coordinates of center of segment I.

CONSTANTS

0.999999	= test for collinear segments

```
      SUBROUTINE CMWW (J,I1,I2,CM,NR,CW,NW,ITRP)
C
C     CMWW COMPUTES MATRIX ELEMENTS FOR WIRE-WIRE INTERACTIONS
C
      COMPLEX CM,CW,ETK,ETS,ETC,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP
     1CON(10),NPCON
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      DIMENSION CM(NR,1), CW(NW,1), CAB(1), SAB(1)
      EQUIVALENCE (CAB,ALP), (SAB,BET)
C     SET SOURCE SEGMENT PARAMETERS
      S=SI(J)
      B=BI(J)
      XJ=X(J)
      YJ=Y(J)
      ZJ=Z(J)
      CABJ=CAB(J)
      SABJ=SAB(J)
      SALPJ=SALP(J)
      IF (IEXK.EQ.0) GO TO 16
C     DECIDE WETHER EXT. T.W. APPROX. CAN BE USED
      IPR=ICON1(J)
      IF (IPR) 1,6,2
1     IPR=-IPR
      IF (-ICON1(IPR).NE.J) GO TO 7
      GO TO 4
2     IF (IPR.NE.J) GO TO 3
      IF (CABJ*CABJ+SABJ*SABJ.GT.1.E-8) GO TO 7
      GO TO 5
3     IF (ICON2(IPR).NE.J) GO TO 7
4     XI=ABS(CABJ*CAB(IPR)+SABJ*SAB(IPR)+SALPJ*SALP(IPR))
      IF (XI.LT.0.999999) GO TO 7
      IF (ABS(BI(IPR)/B-1.).GT.1.E-6) GO TO 7
5     IND1=0
      GO TO 8
6     IND1=1
      GO TO 8
7     IND1=2
8     IPR=ICON2(J)
      IF (IPR) 9,14,10
9     IPR=-IPR
      IF (-ICON2(IPR).NE.J) GO TO 15
      GO TO 12
10    IF (IPR.NE.J) GO TO 11
      IF (CABJ*CABJ+SABJ*SABJ.GT.1.E-8) GO TO 15
      GO TO 13
11    IF (ICON1(IPR).NE.J) GO TO 15
12    XI=ABS(CABJ*CAB(IPR)+SABJ*SAB(IPR)+SALPJ*SALP(IPR))
      IF (XI.LT.0.999999) GO TO 15
      IF (ABS(BI(IPR)/B-1.).GT.1.E-6) GO TO 15
13    IND2=0
      GO TO 16
14    IND2=1
      GO TO 16
15    IND2=2
16    CONTINUE
C
C     OBSERVATION LOOP
C
```

```
      IPR=0
      DO 23 I=I1,I2
      IPR=IPR+1
      IJ=I-J
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      AI=BI(I)
      CABI=CAB(I)
      SABI=SAB(I)
      SALPI=SALP(I)
      CALL EFLD (XI,YI,ZI,AI,IJ)
      ETK=EXK*CABI+EYK*SABI+EZK*SALPI
      ETS=EXS*CABI+EYS*SABI+EZS*SALPI
      ETC=EXC*CABI+EYC*SABI+EZC*SALPI
C
C     FILL MATRIX ELEMENTS.  ELEMENT LOCATIONS DETERMINED BY CONNECTION
C     DATA.
C
      IF (ITRP.NE.0) GO TO 18
C     NORMAL FILL
      DO 17 IJ=1,JSNO
      JX=JCO(IJ)
   17 CM(IPR,JX)=CM(IPR,JX)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
      GO TO 23
   18 IF (ITRP.EQ.2) GO TO 20
C     TRANSPOSED FILL
      DO 19 IJ=1,JSNO
      JX=JCO(IJ)
   19 CM(JX,IPR)=CM(JX,IPR)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
      GO TO 23
C     TRANS. FILL FOR C(WW) - TEST FOR ELEMENTS FOR D(WW)PRIME.  (=CW)
   20 DO 22 IJ=1,JSNO
      JX=JCO(IJ)
      IF (JX.GT.NR) GO TO 21
      CM(JX,IPR)=CM(JX,IPR)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
      GO TO 22
   21 JX=JX-NR
      CW(JX,IPR)=CW(JX,IPR)+ETK*AX(IJ)+ETS*BX(IJ)+ETC*CX(IJ)
   22 CONTINUE
   23 CONTINUE
      RETURN
      END
```

CONECT

PURPOSE

To locate segment ends that contact each other or contact the center of a surface patch.

METHOD

The ends of each segment are identified as end 1 and end 2, defined during geometry input. The connection data for segment I is stored in array variables ICON1 (I) for end 1 and ICON2 (I) for end 2.

Four conditions are possible at each segment end: (1) no connection (a free end), (2) connection to one or more other segments, (3) connection to a ground plane, or (4) connection to a surface modeled with patches. These conditions are indicated in the following way for end 1 of segment I:

(1) no connection . . . . ICON1 (I) = 0

(2) connection to segment J . . . . ICON1 (I) = ±J

(3) connection to a ground plane . . . . . . ICON1 (I) = I

(4) connection to patch K . . . . ICON1 (I) = 10000 + k

In case 2, if segment J has the same reference direction as segment I (end 2 of segment J connected to end 1 of segment I), the sign is positive. For opposed reference directions (end 1 to end 1) the sign is negative. If several segments connect to end 1 of segment I, then J is the number of the next connected segment in sequence.

If segment I connects to patch K, the segment end must coincide with the patch center. Patch K is then divided into four patches numbered K through K + 3 by a call to subroutine SUBPH.

The connection data is illustrated in the following listing for the six segments in the structure in figure 3.

ICON1 (I)	I	ICON2 (I)
10000 + K	1	2
1	2	3
4	3	0
0	4	-5
0	5	6
2	6	0

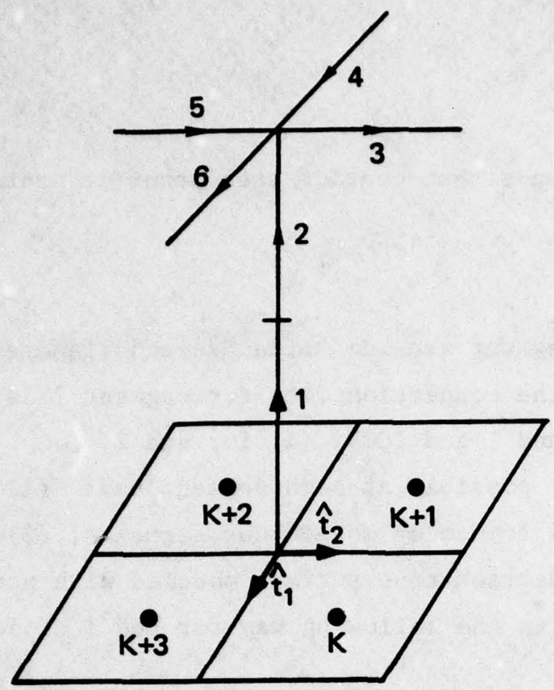


Figure 3. Structure for Illustrating Segment Connection Data.

Connections between patches are not checked, since, except where a wire connects to a surface, the current expansion function on a patch does not extend beyond that patch.

CODING

CN16 - CN27 Initialize and adjust symmetry conditions if necessary when ground is present.

CN40 - CN46 Check whether end 1 of segment I is below ground plane (error) or contacting ground plane. If the separation of the segment end and the ground is less than SMIN multiplied by the segment length, ICON1 is set to I and the z coordinate of the segment end is set to exactly zero.

CN49 - CN60 Check other segments from I + 1 through N and then 1 through I - 1, until a connected end is found. The separation of segment ends is determined by the sum of the separations in x, y, and z to save time.

CN95 - CN126	Search for segments connected to patches. Only new patches (not NGF) are checked. If a connection is found the patch is divided into four patches at its present location in the data arrays and patches following it are shifted up by three locations. This is done by calling SUBPH, an entry point of subroutine PATCH.
CN129 - CN162	Search for new segments connected to NGF patches. If a connection is found four patches, covering the area of the original patch, are added to the end of the data arrays by calling SUBPH. The original patch retains its location but the z coordinate at its center is changed to 10000.
CN182 - CN258	The loop through 44 locates segments connected to junctions.
CN183 - CN190	Parameters are initialized to find all segments connected to first end of segment J.
CN191 - CN215	Connected segments are located. If the number of any connected segment is less than J the loop is exited at CN200. Thus each junction is processed only once.
CN216 - CN230	The connected ends are set to the average of their previous values to ensure that they have identical values.
CN232 - CN244	If the junction includes new segments (NSFLG = 1) and IX is a NGF segment an equation number, NSCON, is assigned for the modified basis function of segment IX. The equation number is stored in array ICONX and the segment number is stored in ISCON.
CN245 - CN247	Segment numbers are printed for junctions of three or more segments.
CN248 - CN257	The loop is initialized for the second end of segment J and the steps from CN191 on are repeated.
CN262 - CN275	Equation numbers for modified basis functions are assigned for old segments that connect to new patches.

SYMBOL DICTIONARY

IGND	=	1 to adjust symmetry for ground and set ICON (I) = I; -1 to adjust symmetry only; 0 for no ground

CONECT

JMAX	=	maximum number of segments connected to a junction
NPMAX	=	maximum number of NGF patches connecting to new segments
NSFLG	=	1 if the junction includes any new segments when NGF is in use
NSMAX	=	maximum number of NGF segments connecting to new segments
SEP	=	approximate separation of segment ends
SLEN	=	maximum separation allowed for connection
SMIN	=	maximum separation as a fraction of segment length
XI1, YI1, ZI1	=	coordinates of end 1 of segment
XI2, YI2, ZI2	=	coordinates of end 2 of segment
XS, YS, ZS	=	coordinates of patch center

CONSTANT

1.E-3	=	maximum separation tolerance for connected segments as fraction of segment length.

```
      SUBROUTINE CONECT (IGND)                                          CN   1
C                                                                       CN   2
C     CONNECT SETS UP SEGMENT CONNECTION DATA IN ARRAYS ICON1 AND ICON2 CN   3
C     BY SEARCHING FOR SEGMENT ENDS THAT ARE IN CONTACT.                CN   4
C                                                                       CN   5
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 CN   6
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( CN   7
     2300),WLAM,IPSYM                                                   CN   8
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP CN   9
     1CON(10),NPCON                                                     CN  10
      DIMENSION X2(1), Y2(1), Z2(1)                                     CN  11
      EQUIVALENCE (X2,SI), (Y2,ALP), (Z2,BET)                           CN  12
      DATA JMAX/30/,SMIN/1.E-3/,NSMAX/50/,NPMAX/10/                     CN  13
      NSCON=0                                                           CN  14
      NPCON=0                                                           CN  15
      IF (IGND.EQ.0) GO TO 3                                            CN  16
      PRINT 54                                                          CN  17
      IF (IGND.GT.0) PRINT 55                                           CN  18
      IF (IPSYM.NE.2) GO TO 1                                           CN  19
      NP=2*NP                                                           CN  20
      MP=2*MP                                                           CN  21
1     IF (IABS(IPSYM).LE.2) GO TO 2                                     CN  22
      NP=N                                                              CN  23
      MP=M                                                              CN  24
2     IF (NP.GT.N) STOP                                                 CN  25
      IF (NP.EQ.N.AND.MP.EQ.M) IPSYM=0                                  CN  26
3     IF (N.EQ.0) GO TO 26                                              CN  27
      DO 15 I=1,N                                                       CN  28
      ICONX(I)=0                                                        CN  29
      XI1=X(I)                                                          CN  30
      YI1=Y(I)                                                          CN  31
      ZI1=Z(I)                                                          CN  32
      XI2=X2(I)                                                         CN  33
      YI2=Y2(I)                                                         CN  34
      ZI2=Z2(I)                                                         CN  35
      SLEN=SQRT((XI2-XI1)**2+(YI2-YI1)**2+(ZI2-ZI1)**2)*SMIN            CN  36
C                                                                       CN  37
C     DETERMINE CONNECTION DATA FOR END 1 OF SEGMENT.                   CN  38
C                                                                       CN  39
      IF (IGND.LT.1) GO TO 5                                            CN  40
      IF (ZI1.GT.-SLEN) GO TO 4                                         CN  41
      PRINT 56, I                                                       CN  42
      STOP                                                              CN  43
4     IF (ZI1.GT.SLEN) GO TO 5                                          CN  44
      ICON1(I)=I                                                        CN  45
      Z(I)=0.                                                           CN  46
      GO TO 9                                                           CN  47
5     IC=I                                                              CN  48
      DO 7 J=2,N                                                        CN  49
      IC=IC+1                                                           CN  50
      IF (IC.GT.N) IC=1                                                 CN  51
      SEP=ABS(XI1-X(IC))+ABS(YI1-Y(IC))+ABS(ZI1-Z(IC))                  CN  52
      IF (SEP.GT.SLEN) GO TO 6                                          CN  53
      ICON1(I)=-IC                                                      CN  54
      GO TO 8                                                           CN  55
6     SEP=ABS(XI1-X2(IC))+ABS(YI1-Y2(IC))+ABS(ZI1-Z2(IC))               CN  56
      IF (SEP.GT.SLEN) GO TO 7                                          CN  57
      ICON1(I)=IC                                                       CN  58
      GO TO 8                                                           CN  59
7     CONTINUE                                                          CN  60
      IF (I.LT.N2.AND.ICON1(I).GT.10000) GO TO 8                        CN  61
      ICON1(I)=0                                                        CN  62
C                                                                       CN  63
C     DETERMINE CONNECTION DATA FOR END 2 OF SEGMENT.                   CN  64
```

```
C                                                                         CN  65
    8 IF (IGND.LT.1) GO TO 12                                             CN  66
    9 IF (ZI2.GT.-SLEN) GO TO 10                                          CN  67
      PRINT 56, I                                                         CN  68
      STOP                                                                CN  69
   10 IF (ZI2.GT.SLEN) GO TO 12                                           CN  70
      IF (ICON1(I).NE.I) GO TO 11                                         CN  71
      PRINT 57, I                                                         CN  72
      STOP                                                                CN  73
   11 ICON2(I)=I                                                          CN  74
      Z2(I)=0.                                                            CN  75
      GO TO 15                                                            CN  76
   12 IC=I                                                                CN  77
      DO 14 J=2,N                                                         CN  78
      IC=IC+1                                                             CN  79
      IF (IC.GT.N) IC=1                                                   CN  80
      SEP=ABS(XI2-X(IC))+ABS(YI2-Y(IC))+ABS(ZI2-Z(IC))                    CN  81
      IF (SEP.GT.SLEN) GO TO 13                                           CN  82
      ICON2(I)=IC                                                         CN  83
      GO TO 15                                                            CN  84
   13 SEP=ABS(XI2-X2(IC))+ABS(YI2-Y2(IC))+ABS(ZI2-Z2(IC))                 CN  85
      IF (SEP.GT.SLEN) GO TO 14                                           CN  86
      ICON2(I)=-IC                                                        CN  87
      GO TO 15                                                            CN  88
   14 CONTINUE                                                            CN  89
      IF (I.LT.N2.AND.ICON2(I).GT.10000) GO TO 15                         CN  90
      ICON2(I)=0                                                          CN  91
   15 CONTINUE                                                            CN  92
      IF (M.EQ.0) GO TO 26                                                CN  93
C     FIND WIRE-SURFACE CONNECTIONS FOR NEW PATCHES                       CN  94
      IX=LD+1-M1                                                          CN  95
      I=M2                                                                CN  96
   16 IF (I.GT.M) GO TO 20                                                CN  97
      IX=IX-1                                                             CN  98
      XS=X(IX)                                                            CN  99
      YS=Y(IX)                                                            CN 100
      ZS=Z(IX)                                                            CN 101
      DO 18 ISEG=1,N                                                      CN 102
      XI1=X(ISEG)                                                         CN 103
      YI1=Y(ISEG)                                                         CN 104
      ZI1=Z(ISEG)                                                         CN 105
      XI2=X2(ISEG)                                                        CN 106
      YI2=Y2(ISEG)                                                        CN 107
      ZI2=Z2(ISEG)                                                        CN 108
      SLEN=(ABS(XI2-XI1)+ABS(YI2-YI1)+ABS(ZI2-ZI1))*SMIN                  CN 109
C     FOR FIRST END OF SEGMENT                                            CN 110
      SEP=ABS(XI1-XS)+ABS(YI1-YS)+ABS(ZI1-ZS)                             CN 111
      IF (SEP.GT.SLEN) GO TO 17                                           CN 112
C     CONNECTION - DIVIDE PATCH INTO 4 PATCHES AT PRESENT ARRAY LOC.      CN 113
      ICON1(ISEG)=10000+I                                                 CN 114
      IC=0                                                                CN 115
      CALL SUBPH (I,IC,XI1,YI1,ZI1,XI2,YI2,ZI2,XA,YA,ZA,XS,YS,ZS)         CN 116
      GO TO 19                                                            CN 117
   17 SEP=ABS(XI2-XS)+ABS(YI2-YS)+ABS(ZI2-ZS)                             CN 118
      IF (SEP.GT.SLEN) GO TO 18                                           CN 119
      ICON2(ISEG)=10000+I                                                 CN 120
      IC=0                                                                CN 121
      CALL SUBPH (I,IC,XI1,YI1,ZI1,XI2,YI2,ZI2,XA,YA,ZA,XS,YS,ZS)         CN 122
      GO TO 19                                                            CN 123
   18 CONTINUE                                                            CN 124
   19 I=I+1                                                               CN 125
      GO TO 16                                                            CN 126
C     REPEAT SEARCH FOR NEW SEGMENTS CONNECTED TO NGF PATCHES.            CN 127
   20 IF (M1.EQ.0.OR.N2.GT.N) GO TO 26                                    CN 128
```

```
      IX=LD+1                                                           CN 129
      I=1                                                               CN 130
   21 IF (I.GT.M1) GO TO 25                                             CN 131
      IX=IX-1                                                           CN 132
      XS=X(IX)                                                          CN 133
      YS=Y(IX)                                                          CN 134
      ZS=Z(IX)                                                          CN 135
      DO 23 ISEG=N2,N                                                   CN 136
      XI1=X(ISEG)                                                       CN 137
      YI1=Y(ISEG)                                                       CN 138
      ZI1=Z(ISEG)                                                       CN 139
      XI2=X2(ISEG)                                                      CN 140
      YI2=Y2(ISEG)                                                      CN 141
      ZI2=Z2(ISEG)                                                      CN 142
      SLEN=(ABS(XI2-XI1)+ABS(YI2-YI1)+ABS(ZI2-ZI1))*SMIN                CN 143
      SEP=ABS(XI1-XS)+ABS(YI1-YS)+ABS(ZI1-ZS)                           CN 144
      IF (SEP.GT.SLEN) GO TO 22                                         CN 145
      ICON1(ISEG)=10001+M                                               CN 146
      IC=1                                                              CN 147
      NPCON=NPCON+1                                                     CN 148
      IPCON(NPCON)=I                                                    CN 149
      CALL SUBPH (I,IC,XI1,YI1,ZI1,XI2,YI2,ZI2,XA,YA,ZA,XS,YS,ZS)       CN 150
      GO TO 24                                                          CN 151
   22 SEP=ABS(XI2-XS)+ABS(YI2-YS)+ABS(ZI2-ZS)                           CN 152
      IF (SEP.GT.SLEN) GO TO 23                                         CN 153
      ICON2(ISEG)=10001+M                                               CN 154
      IC=1                                                              CN 155
      NPCON=NPCON+1                                                     CN 156
      IPCON(NPCON)=I                                                    CN 157
      CALL SUBPH (I,IC,XI1,YI1,ZI1,XI2,YI2,ZI2,XA,YA,ZA,XS,YS,ZS)       CN 158
      GO TO 24                                                          CN 159
   23 CONTINUE                                                          CN 160
   24 I=I+1                                                             CN 161
      GO TO 21                                                          CN 162
   25 IF (NPCON.LE.NPMAX) GO TO 26                                      CN 163
      PRINT 62, NPMAX                                                   CN 164
      STOP                                                              CN 165
   26 PRINT 58, N,NP,IPSYM                                              CN 166
      IF (M.GT.0) PRINT 61, M,MP                                        CN 167
      ISEG=(N+M)/(NP+MP)                                                CN 168
      IF (ISEG.EQ.1) GO TO 30                                           CN 169
      IF (IPSYM) 28,27,29                                               CN 170
   27 STOP                                                              CN 171
   28 PRINT 59, ISEG                                                    CN 172
      GO TO 30                                                          CN 173
   29 IC=ISEG/2                                                         CN 174
      IF (ISEG.EQ.8) IC=3                                               CN 175
      PRINT 60, IC                                                      CN 176
   30 IF (N.EQ.0) GO TO 48                                              CN 177
      PRINT 50                                                          CN 178
      ISEG=0                                                            CN 179
C     ADJUST CONNECTED SEG. ENDS TO EXACTLY COINCIDE.  PRINT JUNCTIONS  CN 180
C     OF 3 OR MORE SEG.  ALSO FIND OLD SEG. CONNECTING TO NEW SEG.      CN 181
      DO 44 J=1,N                                                       CN 182
      IEND=-1                                                           CN 183
      JEND=-1                                                           CN 184
      IX=ICON1(J)                                                       CN 185
      IC=1                                                              CN 186
      JCO(1)=-J                                                         CN 187
      XA=X(J)                                                           CN 188
      YA=Y(J)                                                           CN 189
      ZA=Z(J)                                                           CN 190
   31 IF (IX.EQ.0) GO TO 43                                             CN 191
      IF (IX.EQ.J) GO TO 43                                             CN 192
```

```
      IF (IX.GT.10000) GO TO 43                                        CN 193
      NSFLG=0                                                          CN 194
   32 IF (IX) 33,49,34                                                 CN 195
   33 IX=-IX                                                           CN 196
      GO TO 35                                                         CN 197
   34 JEND=-JEND                                                       CN 198
   35 IF (IX.EQ.J) GO TO 37                                            CN 199
      IF (IX.LT.J) GO TO 43                                            CN 200
      IC=IC+1                                                          CN 201
      IF (IC.GT.JMAX) GO TO 49                                         CN 202
      JCO(IC)=IX*JEND                                                  CN 203
      IF (IX.GT.N1) NSFLG=1                                            CN 204
      IF (JEND.EQ.1) GO TO 36                                          CN 205
      XA=XA+X(IX)                                                      CN 206
      YA=YA+Y(IX)                                                      CN 207
      ZA=ZA+Z(IX)                                                      CN 208
      IX=ICON1(IX)                                                     CN 209
      GO TO 32                                                         CN 210
   36 XA=XA+X2(IX)                                                     CN 211
      YA=YA+Y2(IX)                                                     CN 212
      ZA=ZA+Z2(IX)                                                     CN 213
      IX=ICON2(IX)                                                     CN 214
      GO TO 32                                                         CN 215
   37 SEP=IC                                                           CN 216
      XA=XA/SEP                                                        CN 217
      YA=YA/SEP                                                        CN 218
      ZA=ZA/SEP                                                        CN 219
      DO 39 I=1,IC                                                     CN 220
      IX=JCO(I)                                                        CN 221
      IF (IX.GT.0) GO TO 38                                            CN 222
      IX=-IX                                                           CN 223
      X(IX)=XA                                                         CN 224
      Y(IX)=YA                                                         CN 225
      Z(IX)=ZA                                                         CN 226
      GO TO 39                                                         CN 227
   38 X2(IX)=XA                                                        CN 228
      Y2(IX)=YA                                                        CN 229
      Z2(IX)=ZA                                                        CN 230
   39 CONTINUE                                                         CN 231
      IF (N1.EQ.0) GO TO 42                                            CN 232
      IF (NSFLG.EQ.0) GO TO 42                                         CN 233
      DO 41 I=1,IC                                                     CN 234
      IX=IABS(JCO(I))                                                  CN 235
      IF (IX.GT.N1) GO TO 41                                           CN 236
      IF (ICONX(IX).NE.0) GO TO 41                                     CN 237
      NSCON=NSCON+1                                                    CN 238
      IF (NSCON.LE.NSMAX) GO TO 40                                     CN 239
      PRINT 62, NSMAX                                                  CN 240
      STOP                                                             CN 241
   40 ISCON(NSCON)=IX                                                  CN 242
      ICONX(IX)=NSCON                                                  CN 243
   41 CONTINUE                                                         CN 244
   42 IF (IC.LT.3) GO TO 43                                            CN 245
      ISEG=ISEG+1                                                      CN 246
      PRINT 51, ISEG,(JCO(I),I=1,IC)                                   CN 247
   43 IF (IEND.EQ.1) GO TO 44                                          CN 248
      IEND=1                                                           CN 249
      JEND=1                                                           CN 250
      IX=ICON2(J)                                                      CN 251
      IC=1                                                             CN 252
      JCO(1)=J                                                         CN 253
      XA=X2(J)                                                         CN 254
      YA=Y2(J)                                                         CN 255
      ZA=Z2(J)                                                         CN 256
```

```
          GO TO 31                                                      CN 257
   44     CONTINUE                                                      CN 258
          IF (ISEG.EQ.0) PRINT 52                                       CN 259
          IF (N1.EQ.0.OR.M1.EQ.M) GO TO 48                              CN 260
C         FIND OLD SEGMENTS THAT CONNECT TO NEW PATCHES                 CN 261
          DO 47 J=1,N1                                                  CN 262
          IX=ICON1(J)                                                   CN 263
          IF (IX.LT.10000) GO TO 45                                     CN 264
          IX=IX-10000                                                   CN 265
          IF (IX.GT.M1) GO TO 46                                        CN 266
   45     IX=ICON2(J)                                                   CN 267
          IF (IX.LT.10000) GO TO 47                                     CN 268
          IX=IX-10000                                                   CN 269
          IF (IX.LT.M2) GO TO 47                                        CN 270
   46     IF (ICONX(J).NE.0) GO TO 47                                   CN 271
          NSCON=NSCON+1                                                 CN 272
          ISCON(NSCON)=J                                                CN 273
          ICONX(J)=NSCON                                                CN 274
   47     CONTINUE                                                      CN 275
   48     CONTINUE                                                      CN 276
          RETURN                                                        CN 277
   49     PRINT 53, IX                                                  CN 278
          STOP                                                          CN 279
C                                                                       CN 280
   50     FORMAT (//,9X,27H- MULTIPLE WIRE JUNCTIONS -,/,1X,8HJUNCTION,4X,36 CN 281
         1HSEGMENTS  (- FOR END 1, + FOR END 2))                        CN 282
   51     FORMAT (1X,I5,5X,20I5,/,(11X,20I5))                           CN 283
   52     FORMAT (2X,4HNONE)                                            CN 284
   53     FORMAT (47H CONNECT - SEGMENT CONNECTION ERROR FOR SEGMENT,I5) CN 285
   54     FORMAT (/,3X,23HGROUND PLANE SPECIFIED.)                      CN 286
   55     FORMAT (/,3X,46HWHERE WIRE ENDS TOUCH GROUND, CURRENT WILL BE ,38H CN 287
         1INTERPOLATED TO IMAGE IN GROUND PLANE.,/)                     CN 288
   56     FORMAT (30H GEOMETRY DATA ERROR-- SEGMENT,I5,21H EXTENDS BELOW GRO CN 289
         1UND)                                                          CN 290
   57     FORMAT (29H GEOMETRY DATA ERROR--SEGMENT,I5,16H LIES IN GROUND ,6H CN 291
         1PLANE.)                                                       CN 292
   58     FORMAT (/,3X,20HTOTAL SEGMENTS USED=,I5,5X,12HNO. SEG. IN ,17HA SY CN 293
         1MMETRIC CELL=,I5,5X,14HSYMMETRY FLAG=,I3)                     CN 294
   59     FORMAT (14H STRUCTURE HAS,I4,25H FOLD ROTATIONAL SYMMETRY,/)  CN 295
   60     FORMAT (14H STRUCTURE HAS,I2,19H PLANES OF SYMMETRY,/)        CN 296
   61     FORMAT (3X,19HTOTAL PATCHES USED=,I5,6X,32HNO. PATCHES IN A SYMMET CN 297
         1RIC CELL=,I5)                                                 CN 298
   62     FORMAT (82H ERROR - NO. NEW SEGMENTS CONNECTED TO N.G.F. SEGMENTSO CN 299
         1R PATCHES EXCEEDS LIMIT OF,I5)                                CN 300
          END                                                           CN 301-
```

COUPLE

PURPOSE

To compute the maximum coupling between pairs of segments.

METHOD

If a coupling calculation has been requested (CP card) subroutine COUPLE is called each time that the current is computed for a new excitation. The code from CP10 to CP12 checks that the excitation is a single applied-field voltage source on the segment specified in NCTAG and NCSEG. If the excitation is correct the input admittance and mutual admittances to all other segments specified in NCTAG and NCSEG are stored in Y11A and Y12A from CP13 to CP22.

When all segments have been excited (ICOUP = NCOUP) the second part of the code, from CP24 to CP58 is executed to evaluate the equations in Section V.6 of Part I.

SYMBOL DICTIONARY

C	= L (see Part I, Section V.6)
CUR	= array of values of current at the centers of segments
DBC	= $10 \log(G_{MAX})$
GMAX	= G_{MAX}
ISG1	= segment number
ISG2	= segment number
J1	= index of Y_{12} in array Y12A
J2	= index of Y_{21} in array Y12A
K	= segment number
RHO	= ρ
WLAM	= wavelength
Y11	= Y_{11}
Y12	= $(Y_{12} + Y_{21})/2$
Y22	= Y_{22}
YIN	= Y_{IN}
YL	= Y_L
ZIN	= $1/Y_{IN}$
ZL	= $1/Y_L$

```
      SUBROUTINE COUPLE (CUR,WLAM)                                      CP   1
C                                                                       CP   2
C     COUPLE COMPUTES THE MAXIMUM COUPLING BETWEEN PAIRS OF SEGMENTS.   CP   3
C                                                                       CP   4
      COMPLEX Y11A,Y12A,CUR,Y11,Y12,Y22,YL,YIN,ZL,ZIN,RHO,VQD,VSANT,VQDS CP   5
      COMMON /YPARM/ NCOUP,ICOUP,NCTAG(5),NCSEG(5),Y11A(5),Y12A(20)     CP   6
      COMMON /VSORC/ VQD(30),VSANT(30),VQDS(30),IVQD(30),ISANT(30),IQDS( CP   7
     130),NVQD,NSANT,NQDS                                               CP   8
      DIMENSION CUR(1)                                                  CP   9
      IF (NSANT.NE.1.OR.NVQD.NE.0) RETURN                               CP  10
      J=ISEGNO(NCTAG(ICOUP+1),NCSEG(ICOUP+1))                           CP  11
      IF (J.NE.ISANT(1)) RETURN                                         CP  12
      ICOUP=ICOUP+1                                                     CP  13
      ZIN=VSANT(1)                                                      CP  14
      Y11A(ICOUP)=CUR(J)*WLAM/ZIN                                       CP  15
      L1=(ICOUP-1)*(NCOUP-1)                                            CP  16
      DO 1 I=1,NCOUP                                                    CP  17
      IF (I.EQ.ICOUP) GO TO 1                                           CP  18
      K=ISEGNO(NCTAG(I),NCSEG(I))                                       CP  19
      L1=L1+1                                                           CP  20
      Y12A(L1)=CUR(K)*WLAM/ZIN                                          CP  21
    1 CONTINUE                                                          CP  22
      IF (ICOUP.LT.NCOUP) RETURN                                        CP  23
      PRINT 6                                                           CP  24
      NPM1=NCOUP-1                                                      CP  25
      DO 5 I=1,NPM1                                                     CP  26
      ITT1=NCTAG(I)                                                     CP  27
      ITS1=NCSEG(I)                                                     CP  28
      ISG1=ISEGNO(ITT1,ITS1)                                            CP  29
      L1=I+1                                                            CP  30
      DO 5 J=L1,NCOUP                                                   CP  31
      ITT2=NCTAG(J)                                                     CP  32
      ITS2=NCSEG(J)                                                     CP  33
      ISG2=ISEGNO(ITT2,ITS2)                                            CP  34
      J1=J+(I-1)*NPM1-1                                                 CP  35
      J2=I+(J-1)*NPM1                                                   CP  36
      Y11=Y11A(I)                                                       CP  37
      Y22=Y11A(J)                                                       CP  38
      Y12=.5*(Y12A(J1)+Y12A(J2))                                        CP  39
      YIN=Y12*Y12                                                       CP  40
      DBC=CABS(YIN)                                                     CP  41
      C=DBC/(2.*REAL(Y11)*REAL(Y22)-REAL(YIN))                          CP  42
      IF (C.LT.0..OR.C.GT.1.) GO TO 4                                   CP  43
      IF (C.LT..01) GO TO 2                                             CP  44
      GMAX=(1.-SQRT(1.-C*C))/C                                          CP  45
      GO TO 3                                                           CP  46
    2 GMAX=.5*(C+.25*C*C*C)                                             CP  47
    3 RHO=GMAX*CONJG(YIN)/DBC                                           CP  48
      YL=((1.-RHO)/(1.+RHO)+1.)*REAL(Y22)-Y22                           CP  49
      ZL=1./YL                                                          CP  50
      YIN=Y11-YIN/(Y22+YL)                                              CP  51
      ZIN=1./YIN                                                        CP  52
      DBC=DB10(GMAX)                                                    CP  53
      PRINT 7, ITT1,ITS1,ISG1,ITT2,ITS2,ISG2,DBC,ZL,ZIN                 CP  54
      GO TO 5                                                           CP  55
    4 PRINT 8, ITT1,ITS1,ISG1,ITT2,ITS2,ISG2,C                          CP  56
    5 CONTINUE                                                          CP  57
      RETURN                                                            CP  58
C                                                                       CP  59
    6 FORMAT (///,36X,26H- - - ISOLATION DATA - - -,//,6X,24H- - COUPLIN CP  60
     1G BETWEEN - -,8X,7HMAXIMUM,15X,32H- - - FOR MAXIMUM COUPLING - - - CP  61
     2,/,12X,4HSEG.,14X,4HSEG.,3X,8HCOUPLING,4X,25HLOAD IMPEDANCE (2ND S CP  62
     3EG.),7X,15HINPUT IMPEDANCE,/,2X,8HTAG/SEG.,3X,3HNO.,4X,8HTAG/SEG., CP  63
     43X,3HNO.,6X,4H(DB),8X,4HREAL,9X,5HIMAG.,9X,4HREAL,9X,5HIMAG.)     CP  64
```

```
 7    FORMAT (2(1X,I4,1X,I4,1X,I5,2X),F9.3,2X,2(2X,E12.5,1X,E12.5))      CP  65
 8    FORMAT (2(1X,I4,1X,I4,1X,I5,2X)45H**ERROR** COUPLING IS NOT BETWEE CP  66
     1N 0 AND 1. (=,E12.5,1H))                                           CP  67
      END                                                                CP  68-
```

DATAGN

PURPOSE

To read structure input data and set segment and patch data.

METHOD

The main READ statement is at DA35. The READ statement at DA65 is for the continuation of wire data (GC card following GW), and the READ at DA133 is for the continuation of surface patch data (SC following SP or SM).

The first input parameter GM determines the function of the card as indicated in the following table:

GM	GØ TØ	FUNCTION
GA	8	define wire arc
GC	6	continuation of wire data
GE	29	end of geometry data
GF	27	read NGF file
GM	26	rotate or translate structure
GR	19	rotate about Z axis (symmetry)
GS	21	scale structure
GW	3	define straight wire
GX	18	reflect in coordinate planes (symmetry)
SC	10	continuation of patch data
SM	13	define multiple surface patches
SP	9	define surface patch

The functions of the other input parameters depend on the type of data card and can be determined from the data card descriptions in Part III of this manual.

Subroutines are called to perform many of the operations requested by the data cards. Coding in DATAGN performs other operations, prints information and checks for input errors. After a GE card is read subroutine CONECT is called at DA211 to find electrical connections of segments. Segment and patch data is printed from DA217 to DA256. Line DA241 tests for segments of zero length ($<10^{-20}$) or zero radius ($<10^{-101}$).

SYMBOL DICTIONARY

Variables have multiple uses which depend on the type of input card being processed.

```
      SUBROUTINE DATAGN                                                 DA   1
C                                                                       DA   2
C     DATAGN IS THE MAIN ROUTINE FOR INPUT OF GEOMETRY DATA.            DA   3
C                                                                       DA   4
      INTEGER GM,ATST                                                   DA   5
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 DA   6
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( DA   7
     2300),WLAM,IPSYM                                                   DA   8
      COMMON /ANGL/ SALP(300)                                           DA   9
      DIMENSION X2(1), Y2(1), Z2(1), T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y DA  10
     1(1), T2Z(1), ATST(12), IFX(2), IFY(2), IFZ(2), CAB(1), SAB(1), IPT DA  11
     2(4)                                                               DA  12
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON DA  13
     12), (T2Z,ITAG), (X2,SI), (Y2,ALP), (Z2,BET), (CAB,ALP), (SAB,BET) DA  14
      DATA ATST/2HGW,2HGX,2HGR,2HGS,2HGE,2HGM,2HSP,2HSM,2HGF,2HGA,2HSC,2 DA  15
     1HGC/                                                              DA  16
      DATA IFX/1H ,1HX/,IFY/1H ,1HY/,IFZ/1H ,1HZ/                       DA  17
      DATA TA/0.01745329252/,TD/57.29577951/,IPT/1HP,1HR,1HT,1HQ/       DA  18
      IPSYM=0                                                           DA  19
      NWIRE=0                                                           DA  20
      N=0                                                               DA  21
      NP=0                                                              DA  22
      M=0                                                               DA  23
      MP=0                                                              DA  24
      N1=0                                                              DA  25
      N2=1                                                              DA  26
      M1=0                                                              DA  27
      M2=1                                                              DA  28
      ISCT=0                                                            DA  29
      IPHD=0                                                            DA  30
C                                                                       DA  31
C     READ GEOMETRY DATA CARD AND BRANCH TO SECTION FOR OPERATION       DA  32
C     REQUESTED                                                         DA  33
C                                                                       DA  34
1     READ (5,42) GM,ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD                DA  35
      IF (N+M.GT.LD) GO TO 37                                           DA  36
      IF (GM.EQ.ATST(9)) GO TO 27                                       DA  37
      IF (IPHD.EQ.1) GO TO 2                                            DA  38
      PRINT 40                                                          DA  39
      PRINT 41                                                          DA  40
      IPHD=1                                                            DA  41
2     IF (GM.EQ.ATST(11)) GO TO 10                                      DA  42
      ISCT=0                                                            DA  43
      IF (GM.EQ.ATST(1)) GO TO 3                                        DA  44
      IF (GM.EQ.ATST(2)) GO TO 18                                       DA  45
      IF (GM.EQ.ATST(3)) GO TO 19                                       DA  46
      IF (GM.EQ.ATST(4)) GO TO 21                                       DA  47
      IF (GM.EQ.ATST(7)) GO TO 9                                        DA  48
      IF (GM.EQ.ATST(8)) GO TO 13                                       DA  49
      IF (GM.EQ.ATST(5)) GO TO 29                                       DA  50
      IF (GM.EQ.ATST(6)) GO TO 26                                       DA  51
      IF (GM.EQ.ATST(10)) GO TO 8                                       DA  52
      GO TO 36                                                          DA  53
C                                                                       DA  54
C     GENERATE SEGMENT DATA FOR STRAIGHT WIRE.                          DA  55
C                                                                       DA  56
3     NWIRE=NWIRE+1                                                     DA  57
      I1=N+1                                                            DA  58
      I2=N+NS                                                           DA  59
      PRINT 43, NWIRE,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,NS,I1,I2,ITG          DA  60
      IF (RAD.EQ.0) GO TO 4                                             DA  61
      XS1=1.                                                            DA  62
      YS1=1.                                                            DA  63
      GO TO 7                                                           DA  64
```

```
    4 READ (5,42) GM,IX,IY,XS1,YS1,ZS1
      IF (GM.EQ.ATST(12)) GO TO 6
    5 PRINT 48
      STOP
    6 PRINT 61, XS1,YS1,ZS1
      IF (YS1.EQ.0.OR.ZS1.EQ.0) GO TO 5
      RAD=YS1
      YS1=(ZS1/YS1)**(1./(NS-1.))
    7 CALL WIRE (XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,XS1,YS1,NS,ITG)
      GO TO 1
C
C     GENERATE SEGMENT DATA FOR WIRE ARC
C
    8 NWIRE=NWIRE+1
      I1=N+1
      I2=N+NS
      PRINT 38, NWIRE,XW1,YW1,ZW1,XW2,NS,I1,I2,ITG
      CALL ARC (ITG,NS,XW1,YW1,ZW1,XW2)
      GO TO 1
C
C     GENERATE SINGLE NEW PATCH
C
    9 I1=M+1
      NS=NS+1
      IF (ITG.NE.0) GO TO 17
      PRINT 51, I1,IPT(NS),XW1,YW1,ZW1,XW2,YW2,ZW2
      IF (NS.EQ.2.OR.NS.EQ.4) ISCT=1
      IF (NS.GT.1) GO TO 14
      XW2=XW2*TA
      YW2=YW2*TA
      GO TO 16
   10 IF (ISCT.EQ.0) GO TO 17
      I1=M+1
      NS=NS+1
      IF (ITG.NE.0) GO TO 17
      IF (NS.NE.2.AND.NS.NE.4) GO TO 17
      XS1=X4
      YS1=Y4
      ZS1=Z4
      XS2=X3
      YS2=Y3
      ZS2=Z3
      X3=XW1
      Y3=YW1
      Z3=ZW1
      IF (NS.NE.4) GO TO 11
      X4=XW2
      Y4=YW2
      Z4=ZW2
   11 XW1=XS1
      YW1=YS1
      ZW1=ZS1
      XW2=XS2
      YW2=YS2
      ZW2=ZS2
      IF (NS.EQ.4) GO TO 12
      X4=XW1+X3-XW2
      Y4=YW1+Y3-YW2
      Z4=ZW1+Z3-ZW2
   12 PRINT 51, I1,IPT(NS),XW1,YW1,ZW1,XW2,YW2,ZW2
      PRINT 39, X3,Y3,Z3,X4,Y4,Z4
      GO TO 16
C
C     GENERATE MULTIPLE-PATCH SURFACE
```

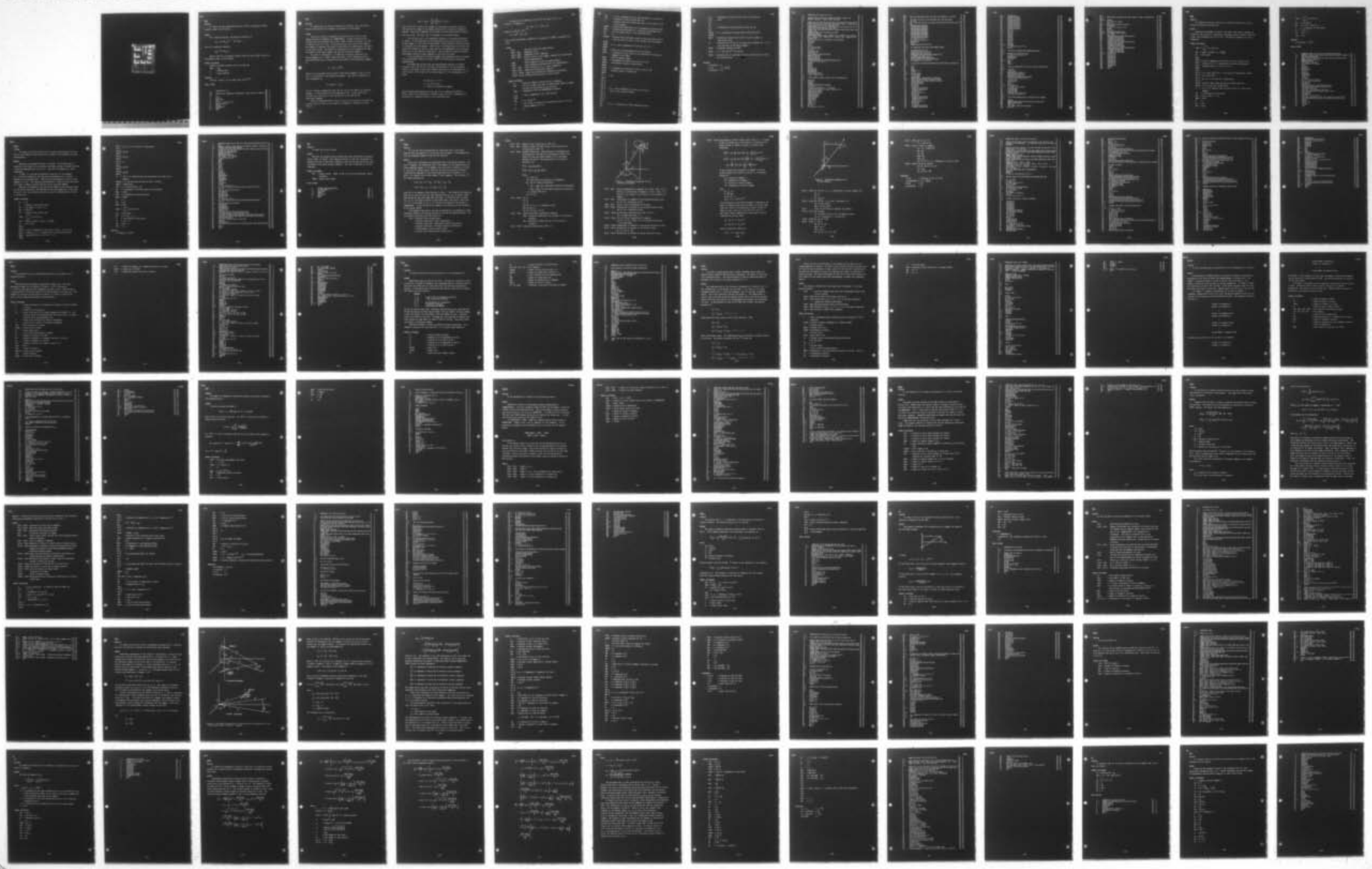
```
      C                                                                         DA 129
      13    I1=M+1                                                              DA 130
            PRINT 59, I1,IPT(2),XW1,YW1,ZW1,XW2,YW2,ZW2,ITG,NS                  DA 131
            IF (ITG.LT.1.OR.NS.LT.1) GO TO 17                                   DA 132
      14    READ (5,42) GM,IX,IY,X3,Y3,Z3,X4,Y4,Z4                              DA 133
            IF (NS.NE.2.AND.ITG.LT.1) GO TO 15                                  DA 134
            X4=XW1+X3-XW2                                                       DA 135
            Y4=YW1+Y3-YW2                                                       DA 136
            Z4=ZW1+Z3-ZW2                                                       DA 137
      15    PRINT 39, X3,Y3,Z3,X4,Y4,Z4                                         DA 138
            IF (GM.NE.ATST(11)) GO TO 17                                        DA 139
      16    CALL PATCH (ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,X3,Y3,Z3,X4,Y4,Z4)       DA 140
            GO TO 1                                                             DA 141
      17    PRINT 60                                                            DA 142
            STOP                                                                DA 143
      C                                                                         DA 144
      C     REFLECT STRUCTURE ALONG X,Y, OR Z AXES OR ROTATE TO FORM CYLINDER.  DA 145
      C                                                                         DA 146
      18    IY=NS/10                                                            DA 147
            IZ=NS-IY*10                                                         DA 148
            IX=IY/10                                                            DA 149
            IY=IY-IX*10                                                         DA 150
            IF (IX.NE.0) IX=1                                                   DA 151
            IF (IY.NE.0) IY=1                                                   DA 152
            IF (IZ.NE.0) IZ=1                                                   DA 153
            PRINT 44, IFX(IX+1),IFY(IY+1),IFZ(IZ+1),ITG                         DA 154
            GO TO 20                                                            DA 155
      19    PRINT 45, NS,ITG                                                    DA 156
            IX=-1                                                               DA 157
      20    CALL REFLC (IX,IY,IZ,ITG,NS)                                        DA 158
            GO TO 1                                                             DA 159
      C                                                                         DA 160
      C     SCALE STRUCTURE DIMENSIONS BY FACTOR XW1.                           DA 161
      C                                                                         DA 162
      21    IF (N.LT.N2) GO TO 23                                               DA 163
            DO 22 I=N2,N                                                        DA 164
            X(I)=X(I)*XW1                                                       DA 165
            Y(I)=Y(I)*XW1                                                       DA 166
            Z(I)=Z(I)*XW1                                                       DA 167
            X2(I)=X2(I)*XW1                                                     DA 168
            Y2(I)=Y2(I)*XW1                                                     DA 169
            Z2(I)=Z2(I)*XW1                                                     DA 170
      22    BI(I)=BI(I)*XW1                                                     DA 171
      23    IF (M.LT.M2) GO TO 25                                               DA 172
            YW1=XW1*XW1                                                         DA 173
            IX=LD+1-M                                                           DA 174
            IY=LD-M1                                                            DA 175
            DO 24 I=IX,IY                                                       DA 176
            X(I)=X(I)*XW1                                                       DA 177
            Y(I)=Y(I)*XW1                                                       DA 178
            Z(I)=Z(I)*XW1                                                       DA 179
      24    BI(I)=BI(I)*YW1                                                     DA 180
      25    PRINT 46, XW1                                                       DA 181
            GO TO 1                                                             DA 182
      C                                                                         DA 183
      C     MOVE STRUCTURE OR REPRODUCE ORIGINAL STRUCTURE IN NEW POSITIONS.    DA 184
      C                                                                         DA 185
      26    PRINT 47, ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD                        DA 186
            XW1=XW1*TA                                                          DA 187
            YW1=YW1*TA                                                          DA 188
            ZW1=ZW1*TA                                                          DA 189
            CALL MOVE (XW1,YW1,ZW1,XW2,YW2,ZW2,INT(RAD+.5),NS,ITG)              DA 190
            GO TO 1                                                             DA 191
      C                                                                         DA 192
```

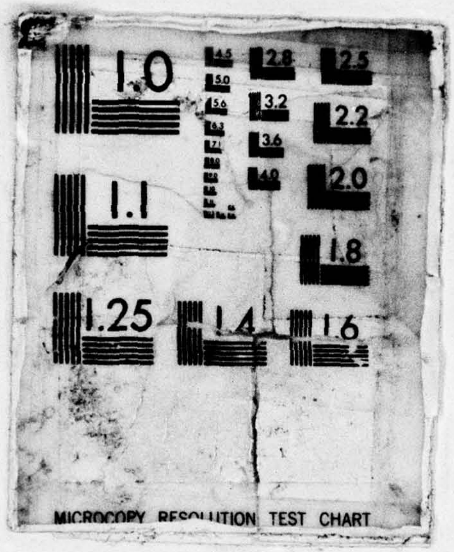
```
C     READ NUMERICAL GREEN'S FUNCTION TAPE
C
27    IF (N+M.EQ.0) GO TO 28
      PRINT 52
      STOP
28    CALL GFIL (ITG)
      NPSAV=NP
      MPSAV=MP
      IPSAV=IPSYM
      GO TO 1
C
C     TERMINATE STRUCTURE GEOMETRY INPUT.
C
29    IX=N1+M1
      IF (IX.EQ.0) GO TO 30
      NP=N
      MP=M
      IPSYM=0
30    CALL CONECT (ITG)
      IF (IX.EQ.0) GO TO 31
      NP=NPSAV
      MP=MPSAV
      IPSYM=IPSAV
31    IF (N+M.GT.LD) GO TO 37
      IF (N.EQ.0) GO TO 33
      PRINT 53
      PRINT 54
      DO 32 I=1,N
      XW1=X2(I)-X(I)
      YW1=Y2(I)-Y(I)
      ZW1=Z2(I)-Z(I)
      X(I)=(X(I)+X2(I))*.5
      Y(I)=(Y(I)+Y2(I))*.5
      Z(I)=(Z(I)+Z2(I))*.5
      XW2=XW1*XW1+YW1*YW1+ZW1*ZW1
      YW2=SQRT(XW2)
      YW2=(XW2/YW2+YW2)*.5
      SI(I)=YW2
      CAB(I)=XW1/YW2
      SAB(I)=YW1/YW2
      XW2=ZW1/YW2
      IF (XW2.GT.1.) XW2=1.
      IF (XW2.LT.-1.) XW2=-1.
      SALP(I)=XW2
      XW2=ASIN(XW2)*TD
      YW2=ATGN2(YW1,XW1)*TD
      PRINT 55, I,X(I),Y(I),Z(I),SI(I),XW2,YW2,BI(I),ICON1(I),I,ICON2(I)
     1,ITAG(I)
      IF (SI(I).GT.1.E-20.AND.BI(I).GT.1.E-101) GO TO 32
      PRINT 56
      STOP
32    CONTINUE
33    IF (M.EQ.0) GO TO 35
      PRINT 57
      J=LD+1
      DO 34 I=1,M
      J=J-1
      XW1=(T1Y(J)*T2Z(J)-T1Z(J)*T2Y(J))*SALP(J)
      YW1=(T1Z(J)*T2X(J)-T1X(J)*T2Z(J))*SALP(J)
      ZW1=(T1X(J)*T2Y(J)-T1Y(J)*T2X(J))*SALP(J)
      PRINT 58, I,X(J),Y(J),Z(J),XW1,YW1,ZW1,BI(J),T1X(J),T1Y(J),T1Z(J),
     1T2X(J),T2Y(J),T2Z(J)
34    CONTINUE
35    RETURN
```

```
36      PRINT 48                                                         DA 257
        PRINT 49, GM,ITG,NS,XW1,YW1,ZW1,XW2,YW2,ZW2,RAD                  DA 258
        STOP                                                             DA 259
37      PRINT 50                                                         DA 260
        STOP                                                             DA 261
C                                                                        DA 262
38      FORMAT (1X,I5,2X,12HARC RADIUS =,F9.5,2X,4HFROM,F8.3,3H TO,F8.3,8H DA 263
       1 DEGREES,11X,F11.5,2X,I5,4X,I5,1X,I5,3X,I5)                      DA 264
39      FORMAT (6X,3F11.5,1X,3F11.5)                                     DA 265
40      FORMAT (////,33X,35H- - - STRUCTURE SPECIFICATION - - -,//,37X,28H DA 266
       1COORDINATES MUST BE INPUT IN,/,37X,29HMETERS OR BE SCALED TO METER DA 267
       2S,/,37X,31HBEFORE STRUCTURE INPUT IS ENDED,//)                   DA 268
41      FORMAT (2X,4HWIRE,79X,6HNO. OF,4X,5HFIRST,2X,4HLAST,5X,3HTAG,/,2X, DA 269
       13HNO.,8X,2HX1,9X,2HY1,9X,2HZ1,10X,2HX2,9X,2HY2,9X,2HZ2,6X,6HRADIUS DA 270
       2,3X,4HSEG.,5X,4HSEG.,3X,4HSEG.,5X,3HNO.)                         DA 271
42      FORMAT (A2,I3,I5,7F10.5)                                         DA 272
43      FORMAT (1X,I5,3F11.5,1X,4F11.5,2X,I5,4X,I5,1X,I5,3X,I5)          DA 273
44      FORMAT (6X,34HSTRUCTURE REFLECTED ALONG THE AXES,3(1X,A1),22H.  TA DA 274
       1GS INCREMENTED BY,I5)                                            DA 275
45      FORMAT (6X,30HSTRUCTURE ROTATED ABOUT Z-AXIS,I3,30H TIMES.  LABLES DA 276
       1 INCREMENTED BY,I5)                                              DA 277
46      FORMAT (6X,26HSTRUCTURE SCALED BY FACTOR,F10.5)                  DA 278
47      FORMAT (6X,49HTHE STRUCTURE HAS BEEN MOVED, MOVE DATA CARD IS -/6X DA 279
       1,I3,I5,7F10.5)                                                   DA 280
48      FORMAT (25H GEOMETRY DATA CARD ERROR)                            DA 281
49      FORMAT (1X,A2,I3,I5,7F10.5)                                      DA 282
50      FORMAT (69H NUMBER OF WIRE SEGMENTS AND SURFACE PATCHES EXCEEDS DI DA 283
       1MENSION LIMIT.)                                                  DA 284
51      FORMAT (1X,I5,A1,F10.5,2F11.5,1X,3F11.5)                         DA 285
52      FORMAT (44H ERROR - GF MUST BE FIRST GEOMETRY DATA CARD)         DA 286
53      FORMAT (////33X,33H- - - - SEGMENTATION DATA - - - -,//,40X,21HCOO DA 287
       1RDINATES IN METERS,//,25X,50HI+ AND I- INDICATE THE SEGMENTS BEFOR DA 288
       2E AND AFTER I,//)                                                DA 289
54      FORMAT (2X,4HSEG.,3X,26HCOORDINATES OF SEG. CENTER,5X,4HSEG.,5X,18 DA 290
       1HORIENTATION ANGLES,4X,4HWIRE,4X,15HCONNECTION DATA,3X,3HTAG,/,2X, DA 291
       23HNO.,7X,1HX,9X,1HY,9X,1HZ,7X,6HLENGTH,5X,5HALPHA,5X,4HBETA,6X,6HR DA 292
       3ADIUS,4X,2HI-,3X,1HI,4X,2HI+,4X,3HNO.)                           DA 293
55      FORMAT (1X,I5,4F10.5,1X,3F10.5,1X,3I5,2X,I5)                     DA 294
56      FORMAT (19H SEGMENT DATA ERROR)                                  DA 295
57      FORMAT (////,44X,30H- - - SURFACE PATCH DATA - - -,//,49X,21HCOORD DA 296
       1INATES IN METERS,//,1X,5HPATCH,5X,22HCOORD. OF PATCH CENTER,7X,18H DA 297
       2UNIT NORMAL VECTOR,6X,5HPATCH,12X,34HCOMPONENTS OF UNIT TANGENT VE DA 298
       3CTORS,/,2X,3HNO.,6X,1HX,9X,1HY,9X,1HZ,9X,1HX,7X,1HY,7X,1HZ,7X,4HAR DA 299
       4EA,7X,2HX1,6X,2HY1,6X,2HZ1,7X,2HX2,6X,2HY2,6X,2HZ2)              DA 300
58      FORMAT (1X,I4,3F10.5,1X,3F8.4,F10.5,1X,3F8.4,1X,3F8.4)           DA 301
59      FORMAT (1X,I5,A1,F10.5,2F11.5,1X,3F11.5,5X,9HSURFACE -,I4,3H BY,I3 DA 302
       1,8H PATCHES)                                                     DA 303
60      FORMAT (17H PATCH DATA ERROR)                                    DA 304
61      FORMAT (9X,43HABOVE WIRE IS TAPERED.  SEG. LENGTH RATIO =,F9.5,/,3 DA 305
       13X,11HRADIUS FROM,F9.5,3H TO,F9.5)                               DA 306
        END                                                              DA 307-
```

AD-A075 289 NUMERICAL ELECTROMAGNETIC CODE (NEC)-METHOD OF MOMENTS 3/6
A USER-ORIENTED CO..(U) NAVAL OCEAN SYSTEMS CENTER SAN
DIEGO CA G J BURKE ET AL. 18 JUL 77 NOSC/TD-116-VOL-1

UNCLASSIFIED AFWL-TR-76-320-VOL-1 F/G 12/5 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

DB10

PURPOSE

To convert an input magnitude quantity (field) or magnitude squared quantity (power) into decibels.

METHOD

For a squared quantity, the decibel conversion is

$$Q_{db} = 10 \log_{10} Q^2 \qquad (Q^2 \text{ input}) ,$$

and for an unsquared quantity,

$$Q_{db} = 20 \log_{10} Q .$$

DB10 is used for the squared quantity while the entry DB20 is used for the quantity which is not squared.

SYMBOL DICTIONARY

ALOG10 = external routine (log to the base 10)

DB10 = Q_{db}

F = scaling term

X = input quantity

CONSTANT

-999.99 = returned for an input less than 10^{-20}

CODE LISTING

```
      FUNCTION DB10 (X)                                                  DB   1
C                                                                        DB   2
C     FUNCTION DB-- RETURNS DB FOR MAGNITUDE (FIELD) OR MAG**2 (POWER) I DB   3
C                                                                        DB   4
      F=10.                                                              DB   5
      GO TO 1                                                            DB   6
      ENTRY DB20                                                         DB   7
      F=20.                                                              DB   8
    1 IF (X.LT.1.E-20) GO TO 2                                           DB   9
      DB10=F*ALOG10(X)                                                   DB  10
      RETURN                                                             DB  11
    2 DB10=-999.99                                                       DB  12
      RETURN                                                             DB  13
      END                                                                DB  14-
```

EFLD

PURPOSE

To compute the near electric field due to constant, sine, and cosine current distributions on a segment in free space or over ground.

METHOD

The electric field is computed at the point XI, YI, ZI due to the segment defined by parameters in COMMON/DATAJ/. Either the thin wire or extended thin wire formulas may be used. When a ground is present, the code is executed twice in a loop. In the second pass, the field of the image of the segment is computed, multiplied by the reflection coefficients, and added to the direct field. The reflection coefficients for the reflected ray from the center of the source segment are used for the entire segment.

The field is evaluated in a cylindrical coordinate system with the source segment at the origin, along the z axis. The ρ coordinate of the field evaluation point is computed for the surface of the observation segment as

$$\rho' = (\rho^2 + a^2)^{1/2},$$

where ρ is the distance from the axis of the source segment to (XI, YI, ZI) and a is the radius of the observation segment. The field is computed in ρ and z components as

$$\bar{E} = E_\rho(\bar{\rho}/\rho') + E_z\hat{z} \, .$$

Use of ρ' avoids a singularity when (XI, YI, ZI) is the center of the source segment. In the addition of field components, $\bar{\rho}/\rho'$ is used rather than $\hat{\rho}$, since E_ρ is the field in the direction $\hat{\rho}'$ to one side of the observation segment.

When the Sommerfeld/Norton option is used for an antenna over ground the electric field at $\bar{r}$ due to the current on a segment is evaluated in three terms as

EFLD

$$\bar{E}(\bar{r}) = \bar{E}_D(\bar{r}) + \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2}\bar{E}_I(\bar{r}) + \bar{E}_S(\bar{r})$$

$\bar{E}_D$ is the direct field of the segment in the absence of ground, and $\bar{E}_I$ is the field of the image of the segment reflected in a perfectly conducting ground. These field comonents are evaluated in EFLD between EF19 and EF150. The factor $(k_1^2 - k_2^2)/(k_1^2 + k_2^2)$ is contained in the variable FRATI.

The field $\bar{E}_S$, due to the Sommerfeld integrals is evaluated from EF155 to EF227. If the separation of the observation point and the center of the source segment is less than one wavelength, subroutine ROM2 is called at EF191 to integrate over the segment. DMIN is set to the magnitude of the first two terms in $\bar{E}$ divided by 100 as a lower limit on the denominator of the relative error test in the numerical integration. This relaxes the relative accuracy requirement when $\bar{E}_S$ is small compared to the first two terms.

If the separation of the source segment and observation point is greater than a wavelength, SFLDS is called at EF197 to evaluate $\bar{E}_S$ by the Norton approximation.

To compute $\bar{E}_S$ with the thin wire approximation applied in a manner consistent with that for $\bar{E}_I$, the field is evaluated at a point displaced normal to the image of the source segment and normal to the separation $\bar{R}$. If the direction of the image of the source segment is $\hat{j}$ the displacement is $\bar{D}$ where

$$\bar{D} = \pm a\hat{d} \text{ for } \hat{z} \cdot \hat{d} \gtrless 0$$

$$\hat{d} = (\hat{j} \times \bar{R})/|\hat{j} \times \bar{R}|$$

$$a = \text{radius of observation segment}$$

This displaced observation point (XO, YO, ZO) is computed from EF166 to EF181. Some of the complexity is needed to make the result independent of orientation of segments relative to the coordinate axes.

To adjust the ρ component of field for the factor $|\bar{\rho}/\rho'|$ the field $\bar{E}'$ is computed as

$$\bar{E}' = F\bar{E} + (1 - F)(\bar{E} \cdot \hat{j})\hat{j}$$

where $F = [\rho^2/(\rho^2 + a^2)]^{1/2}$

$\rho^2 = |\bar{R}|^2 - (\bar{R} \cdot \hat{j})^2$

This is done from EF204 to EF218 but is skipped if F (DMIN) is greater than 0.95.

CODING

EF23	Loop over direct and image fields.
EF29 - EF31	Components of $\bar{\rho}$.
EF33 - EF40	Components of $\bar{\rho}/\rho'$ computed.
EF46 - EF62	Electric field of the segment computed by infinitesimal dipole approximation.
EF68	Field computed by thin wire approximation.
EF70	Field computed by extended thin wire approximation.
EF72 - EF80	Field converted to x, y, and z components.
EF89 - EF111	Reflection coefficients computed.
EF112 - EF129	Image fields modified by reflection coefficients.
EF130 - EF138	Reflected fields added to direct fields.

SYMBOL DICTIONARY

AI	= radius of segment on which field is evaluated
CTH	= cos θ; θ = angle from axis of infinitesimal dipole or angle between the reflecting ray and vertical
EGND	= components of $\bar{E}_S$ (see EQUIVALENCE statement)
EPX, EPY	= x and y components of $(\bar{E} \cdot \hat{p})\hat{p}$ (see PX)
ETA	= $\eta = (\mu_o/\varepsilon_o)^{1/2}$
IJ	= IJX = flag to indicate field evaluation point is on the source segment (IJ = 0)
PI	= π

PX, PY	= x and y components of unit vector normal to the plane of incidence of the reflected wave ($\hat{p}$)
R	= distance from field evaluation point to the center of the source segment
REFPS	= reflection coefficient for a horizontally polarized field
REFS	= reflection coefficient for a vertically polarized field
RFL	= +1 for direct field, -1 for reflected field
RH	= ρ'
RHOSPC	= distance from coordinate origin to the point where the ray from the source to (XI, YI, ZI) reflects from the ground
RHOX, RHOY, RHOZ	= x, y, and z components of $\bar{\rho}$ or $\bar{\rho}/\rho'$ or $\hat{j} \times \bar{R}$
RMAG	= 2πR or R or dipole moment for sin ks current
SALPR	= z component of unit vector in the direction of the source segment or its image
SHAF	= half of segment length
TERC, TERS, TERK	= ρ component of field due to cos ks, sin ks, and constant currents, respectively
TEZC, TEZS, TEZK	= z component of field due to cos ks, sin ks, and constant current, respectively
TP	= 2π
TXC, TYC, TZC, TXS, TYS, TZS, TXK, TYK, TZK	= x, y, and z components of field due to cos ks, sin ks, and constant current
XI, YI, ZI	= x, y, z coordinates of field evaluation point

XIJ, YIJ, ZIJ = components of distance from source to observation point

XO, YO, ZO = coordinates of field evaluation point for E_S

XSPEC, YSPEC = x, y coordinates of ground plane reflection point

XYMAG = horizontal distance from center of source segment to observation point

ZP = projection of the vector from the source segment (XI, YI, ZI) onto the axis of the source segment

ZRATX = temporary storage for ZRATI

ZRSIN = $(1 - Z_R^2 \sin^2 \theta)^{1/2}$ for ground

ZSCRN = quantity used in computing reflection coefficient for radial wire ground screen

CONSTANT

3.141592654 = π

376.73 = $\eta = \sqrt{\mu_0/\epsilon_0}$

6.283185308 = 2π

```
      SUBROUTINE EFLD (XI,YI,ZI,AI,IJ)
C
C     COMPUTE NEAR E FIELDS OF A SEGMENT WITH SINE, COSINE, AND
C     CONSTANT CURRENTS.  GROUND EFFECT INCLUDED.
C
      COMPLEX TXK,TYK,TZK,TXS,TYS,TZS,TXC,TYC,TZC,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,EPX,EPY,ZRATI,REFS,REFPS,ZRSIN,ZRATX,T1,ZSCRN,ZRATI2
     2,TEZS,TERS,TEZC,TERC,TEZK,TERK,EGND,FRATI
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      COMMON /GND/ ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR
     1,IPERF,T1,T2
      COMMON /INCOM/ XO,YO,ZO,SN,XSN,YSN,ISNOR
      DIMENSION EGND(9)
      EQUIVALENCE (EGND(1),TXK), (EGND(2),TYK), (EGND(3),TZK), (EGND(4),
     1TXS), (EGND(5),TYS), (EGND(6),TZS), (EGND(7),TXC), (EGND(8),TYC),
     2(EGND(9),TZC)
      DATA ETA/376.73/,PI/3.141592654/,TP/6.283185308/
      XIJ=XI-XJ
      YIJ=YI-YJ
      IJX=IJ
      RFL=-1.
      DO 12 IP=1,KSYMP
      IF (IP.EQ.2) IJX=1
      RFL=-RFL
      SALPR=SALPJ*RFL
      ZIJ=ZI-RFL*ZJ
      ZP=XIJ*CABJ+YIJ*SABJ+ZIJ*SALPR
      RHOX=XIJ-CABJ*ZP
      RHOY=YIJ-SABJ*ZP
      RHOZ=ZIJ-SALPR*ZP
      RH=SQRT(RHOX*RHOX+RHOY*RHOY+RHOZ*RHOZ+AI*AI)
      IF (RH.GT.1.E-10) GO TO 1
      RHOX=0.
      RHOY=0.
      RHOZ=0.
      GO TO 2
1     RHOX=RHOX/RH
      RHOY=RHOY/RH
      RHOZ=RHOZ/RH
2     R=SQRT(ZP*ZP+RH*RH)
      IF (R.LT.RKH) GO TO 3
C
C     LUMPED CURRENT ELEMENT APPROX. FOR LARGE SEPARATIONS
C
      RMAG=TP*R
      CTH=ZP/R
      PX=RH/R
      TXK=CMPLX(COS(RMAG),-SIN(RMAG))
      PY=TP*R*R
      TYK=ETA*CTH*TXK*CMPLX(1.,-1./RMAG)/PY
      TZK=ETA*PX*TXK*CMPLX(1.,RMAG-1./RMAG)/(2.*PY)
      TEZK=TYK*CTH-TZK*PX
      TERK=TYK*PX+TZK*CTH
      RMAG=SIN(PI*S)/PI
      TEZC=TEZK*RMAG
      TERC=TERK*RMAG
      TEZK=TEZK*S
      TERK=TERK*S
      TXS=(0.,0.)
      TYS=(0.,0.)
      TZS=(0.,0.)
      GO TO 6
3     IF (IEXK.EQ.1) GO TO 4
```

```
C
C     EKSC FOR THIN WIRE APPROX. OR EKSCX FOR EXTENDED T.W. APPROX.
C
      CALL EKSC (S,ZP,RH,TP,IJX,TEZS,TERS,TEZC,TERC,TEZK,TERK)
      GO TO 5
4     CALL EKSCX (B,S,ZP,RH,TP,IJX,IND1,IND2,TEZS,TERS,TEZC,TERC,TEZK,TE
     1RK)
5     TXS=TEZS*CABJ+TERS*RHOX
      TYS=TEZS*SABJ+TERS*RHOY
      TZS=TEZS*SALPR+TERS*RHOZ
6     TXK=TEZK*CABJ+TERK*RHOX
      TYK=TEZK*SABJ+TERK*RHOY
      TZK=TEZK*SALPR+TERK*RHOZ
      TXC=TEZC*CABJ+TERC*RHOX
      TYC=TEZC*SABJ+TERC*RHOY
      TZC=TEZC*SALPR+TERC*RHOZ
      IF (IP.NE.2) GO TO 11
      IF (IPERF.GT.0) GO TO 10
      ZRATX=ZRATI
      RMAG=R
      XYMAG=SQRT(XIJ*XIJ+YIJ*YIJ)
C
C     SET PARAMETERS FOR RADIAL WIRE GROUND SCREEN.
C
      IF (NRADL.EQ.0) GO TO 7
      XSPEC=(XI*ZJ+ZI*XJ)/(ZI+ZJ)
      YSPEC=(YI*ZJ+ZI*YJ)/(ZI+ZJ)
      RHOSPC=SQRT(XSPEC*XSPEC+YSPEC*YSPEC+T2*T2)
      IF (RHOSPC.GT.SCRWL) GO TO 7
      ZSCRN=T1*RHOSPC*ALOG(RHOSPC/T2)
      ZRATX=(ZSCRN*ZRATI)/(ETA*ZRATI+ZSCRN)
7     IF (XYMAG.GT.1.E-6) GO TO 8
C
C     CALCULATION OF REFLECTION COEFFICIENTS WHEN GROUND IS SPECIFIED.
C
      PX=0.
      PY=0.
      CTH=1.
      ZRSIN=(1.,0.)
      GO TO 9
8     PX=-YIJ/XYMAG
      PY=XIJ/XYMAG
      CTH=ZIJ/RMAG
      ZRSIN=CSQRT(1.-ZRATX*ZRATX*(1.-CTH*CTH))
9     REFS=(CTH-ZRATX*ZRSIN)/(CTH+ZRATX*ZRSIN)
      REFPS=-(ZRATX*CTH-ZRSIN)/(ZRATX*CTH+ZRSIN)
      REFPS=REFPS-REFS
      EPY=PX*TXK+PY*TYK
      EPX=PX*EPY
      EPY=PY*EPY
      TXK=REFS*TXK+REFPS*EPX
      TYK=REFS*TYK+REFPS*EPY
      TZK=REFS*TZK
      EPY=PX*TXS+PY*TYS
      EPX=PX*EPY
      EPY=PY*EPY
      TXS=REFS*TXS+REFPS*EPX
      TYS=REFS*TYS+REFPS*EPY
      TZS=REFS*TZS
      EPY=PX*TXC+PY*TYC
      EPX=PX*EPY
      EPY=PY*EPY
      TXC=REFS*TXC+REFPS*EPX
      TYC=REFS*TYC+REFPS*EPY
```

```
      TZC=REFS*TZC
   10 EXK=EXK-TXK*FRATI
      EYK=EYK-TYK*FRATI
      EZK=EZK-TZK*FRATI
      EXS=EXS-TXS*FRATI
      EYS=EYS-TYS*FRATI
      EZS=EZS-TZS*FRATI
      EXC=EXC-TXC*FRATI
      EYC=EYC-TYC*FRATI
      EZC=EZC-TZC*FRATI
      GO TO 12
   11 EXK=TXK
      EYK=TYK
      EZK=TZK
      EXS=TXS
      EYS=TYS
      EZS=TZS
      EXC=TXC
      EYC=TYC
      EZC=TZC
   12 CONTINUE
      IF (IPERF.EQ.2) GO TO 13
      RETURN
C
C     FIELD DUE TO GROUND USING SOMMERFELD/NORTON
C
   13 SN=SQRT(CABJ*CABJ+SABJ*SABJ)
      IF (SN.LT.1.E-5) GO TO 14
      XSN=CABJ/SN
      YSN=SABJ/SN
      GO TO 15
   14 SN=0.
      XSN=1.
      YSN=0.
C
C     DISPLACE OBSERVATION POINT FOR THIN WIRE APPROXIMATION
C
   15 ZIJ=ZI+ZJ
      SALPR=-SALPJ
      RHOX=SABJ*ZIJ-SALPR*YIJ
      RHOY=SALPR*XIJ-CABJ*ZIJ
      RHOZ=CABJ*YIJ-SABJ*XIJ
      RH=RHOX*RHOX+RHOY*RHOY+RHOZ*RHOZ
      IF (RH.GT.1.E-10) GO TO 16
      XO=XI-AI*YSN
      YO=YI+AI*XSN
      ZO=ZI
      GO TO 17
   16 RH=AI/SQRT(RH)
      IF (RHOZ.LT.0.) RH=-RH
      XO=XI+RH*RHOX
      YO=YI+RH*RHOY
      ZO=ZI+RH*RHOZ
   17 R=XIJ*XIJ+YIJ*YIJ+ZIJ*ZIJ
      IF (R.GT..95) GO TO 18
C
C     FIELD FROM INTERPOLATION IS INTEGRATED OVER SEGMENT
C
      ISNOR=1
      DMIN=EXK*CONJG(EXK)+EYK*CONJG(EYK)+EZK*CONJG(EZK)
      DMIN=.01*SQRT(DMIN)
      SHAF=.5*S
      CALL ROM2 (-SHAF,SHAF,EGND,DMIN)
      GO TO 19
```

```
C                                                                    EF 193
C     NORTON FIELD EQUATIONS AND LUMPED CURRENT ELEMENT APPROXIMATION  EF 194
C                                                                    EF 195
18    ISNOR=2                                                        EF 196
      CALL SFLDS (0.,EGND)                                           EF 197
      GO TO 22                                                       EF 198
19    ZP=XIJ*CABJ+YIJ*SABJ+ZIJ*SALPR                                 EF 199
      RH=R-ZP*ZP                                                     EF 200
      IF (RH.GT.1.E-10) GO TO 20                                     EF 201
      DMIN=0.                                                        EF 202
      GO TO 21                                                       EF 203
20    DMIN=SQRT(RH/(RH+AI*AI))                                       EF 204
21    IF (DMIN.GT..95) GO TO 22                                      EF 205
      PX=1.-DMIN                                                     EF 206
      TERK=(TXK*CABJ+TYK*SABJ+TZK*SALPR)*PX                          EF 207
      TXK=DMIN*TXK+TERK*CABJ                                         EF 208
      TYK=DMIN*TYK+TERK*SABJ                                         EF 209
      TZK=DMIN*TZK+TERK*SALPR                                        EF 210
      TERS=(TXS*CABJ+TYS*SABJ+TZS*SALPR)*PX                          EF 211
      TXS=DMIN*TXS+TERS*CABJ                                         EF 212
      TYS=DMIN*TYS+TERS*SABJ                                         EF 213
      TZS=DMIN*TZS+TERS*SALPR                                        EF 214
      TERC=(TXC*CABJ+TYC*SABJ+TZC*SALPR)*PX                          EF 215
      TXC=DMIN*TXC+TERC*CABJ                                         EF 216
      TYC=DMIN*TYC+TERC*SABJ                                         EF 217
      TZC=DMIN*TZC+TERC*SALPR                                        EF 218
22    EXK=EXK+TXK                                                    EF 219
      EYK=EYK+TYK                                                    EF 220
      EZK=EZK+TZK                                                    EF 221
      EXS=EXS+TXS                                                    EF 222
      EYS=EYS+TYS                                                    EF 223
      EZS=EZS+TZS                                                    EF 224
      EXC=EXC+TXC                                                    EF 225
      EYC=EYC+TYC                                                    EF 226
      EZC=EZC+TZC                                                    EF 227
      RETURN                                                         EF 228
      END                                                            EF 229-
```

EKSC

PURPOSE

To compute the electric field due to current filaments with sin kz, cos kz and constant distributions.

METHOD

Equations 71 through 74 in Part I are used. The current filament is located at the origin of a cylindrical coordinate system, oriented along the z axis, and extending from $-\Delta/2$ to $\Delta/2$. The field is computed in ρ and z components.

SYMBOL DICTIONARY

CINT = $\int_{-\Delta/2}^{\Delta/2} \cos(kr)/r \, dz$

CON = CONX = $j\eta/(8\pi^2)$, $\eta = \sqrt{\mu_0/\varepsilon_0}$

CS = $\cos(k\Delta/2)$

ERS, EZS, ERC, EZC, ERK, EZK = ρ and z components of field due to sin kz, cos kz, and constant (S, C, K, respectively) current distributions extending from $z = -\Delta/2$ to $z = \Delta/2$

GP1, GP2 = $-(1 + jkr)\, G_0/r^2$ for $z = -\Delta/2$ and $\Delta/2$, respectively, where $G_0 = \exp(-jkr)/r$

GZ1, GZ2 = G_0 for $z = -\Delta/2$ and $\Delta/2$, respectively

GZP1, GZP2 = $\partial G_0/\partial z$ at EK21, EK22 and $\partial G_0/\partial \rho$ at EK28, EK29 for $z = -\Delta/2$ and $\Delta/2$, respectively

IJ = IJX = 0 to indicate that the field point is on the source segment

RH = ρ coordinate of field point

RHK = $k\rho$ ($k = 2\pi/\lambda$, $\lambda = 1$)

RKB2 = $(k\rho)^2$

S = Δ

SH = $\Delta/2$

SHK = $k\Delta/2$

$$\text{SINT} = \int_{-\Delta/2}^{\Delta/2} \sin(kr)/r \, dz$$

SS = $\sin(k\Delta/2)$

XK = $k = 2\pi/\lambda$, where $\lambda = 1$

Z = z coordinate of field point

Z1 = $-\Delta/2 - z$

Z2 = $\Delta/2 - z$

ZPK = kz

CONSTANT

$4.771341189 = \eta/(8\pi^2)$

CODE LISTING

```
      SUBROUTINE EKSC (S,Z,RH,XK,IJ,EZS,ERS,EZC,ERC,EZK,ERK)             EK   1
C     COMPUTE E FIELD OF SINE, COSINE, AND CONSTANT CURRENT FILAMENTS BY EK   2
C     THIN WIRE APPROXIMATION.                                           EK   3
      COMPLEX CON,GZ1,GZ2,GP1,GP2,GZP1,GZP2,EZS,ERS,EZC,ERC,EZK,ERK      EK   4
      COMMON /TMI/ ZPK,RKB2,IJX                                          EK   5
      DIMENSION CONX(2)                                                  EK   6
      EQUIVALENCE (CONX,CON)                                             EK   7
      DATA CONX/0.,4.771341189/                                          EK   8
      IJX=IJ                                                             EK   9
      ZPK=XK*Z                                                           EK  10
      RHK=XK*RH                                                          EK  11
      RKB2=RHK*RHK                                                       EK  12
      SH=.5*S                                                            EK  13
      SHK=XK*SH                                                          EK  14
      SS=SIN(SHK)                                                        EK  15
      CS=COS(SHK)                                                        EK  16
      Z2=SH-Z                                                            EK  17
      Z1=-(SH+Z)                                                         EK  18
      CALL GX (Z1,RH,XK,GZ1,GP1)                                         EK  19
      CALL GX (Z2,RH,XK,GZ2,GP2)                                         EK  20
      GZP1=GP1*Z1                                                        EK  21
      GZP2=GP2*Z2                                                        EK  22
      EZS=CON*((GZ2-GZ1)*CS*XK-(GZP2+GZP1)*SS)                           EK  23
      EZC=-CON*((GZ2+GZ1)*SS*XK+(GZP2-GZP1)*CS)                          EK  24
      ERK=CON*(GP2-GP1)*RH                                               EK  25
      CALL INTX (-SHK,SHK,RHK,IJ,CINT,SINT)                              EK  26
      EZK=-CON*(GZP2-GZP1+XK*XK*CMPLX(CINT,-SINT))                       EK  27
      GZP1=GZP1*Z1                                                       EK  28
      GZP2=GZP2*Z2                                                       EK  29
      IF (RH.LT.1.E-10) GO TO 1                                          EK  30
      ERS=-CON*((GZP2+GZP1+GZ2+GZ1)*SS-(Z2*GZ2-Z1*GZ1)*CS*XK)/RH         EK  31
      ERC=-CON*((GZP2-GZP1+GZ2-GZ1)*CS+(Z2*GZ2+Z1*GZ1)*SS*XK)/RH         EK  32
      RETURN                                                             EK  33
1     ERS=(0.,0.)                                                        EK  34
      ERC=(0.,0.)                                                        EK  35
      RETURN                                                             EK  36
      END                                                                EK  37-
```

EKSCX

PURPOSE

To compute the electric field due to current distributions of sin kz, cos kz, and constant on the surface of a cylinder by the extended thin wire approximation.

METHOD

Equations 84 through 87 in Part I are used. The current tube is centered on the origin of a cylindrical coordinate system, oriented along the z axis and extending from $-\Delta/2$ to $\Delta/2$. The field is computed in ρ and z components.

If INX1 = 2, the field contributions from end 1 of the segment ($z = -\Delta/2$) are evaluated by the thin wire approximation for a current filament on the cylinder axis. INX2 has the same meaning for end 2 of the segment ($z = \Delta/2$). The thin-wire approximation is used at an end when there is a bend or change in radius from that end to the next segment.

When the ρ coordinate of the field point (RHX) is less than the radius of the current tube (BX), then RHX and BX are interchanged and a flag, IRA, is set to 1 to cause alternate forms for G_1 and its derivatives to be used in routine GXX.

SYMBOL DICTIONARY

A2 = B^2

B = radius of the current tube

BK = kB, where $k = 2\pi/\lambda$, $\lambda = 1$

BK2 = $(BK)^2/4$

BX = radius of the current tube

CINT = $\int_{-\Delta/2}^{\Delta/2} \cos(kr)/r \, dz$

CON = CONX = $j\eta/(8\pi^2)$, where $\eta = \sqrt{\mu_0/\varepsilon_0}$

CS = $\cos(k\Delta/2)$

$\left.\begin{array}{l}\text{ERS}\\ \text{EZS}\\ \text{ERC}\\ \text{EZC}\\ \text{ERK}\\ \text{EZK}\end{array}\right\}$ = ρ and z components of field due to sin kz, cos kz, and constant (S, C, K, respectively) current distributions extending from $z = -\Delta/2$ to $z = \Delta/2$.

GR1, GR2	$= G_2$ for $z = -\Delta/2$ and $\Delta/2$, respectively
GRK1, GRK2	$= \partial G_1/\partial\rho$
GRP1, GRP2	$= \partial G_2/\partial z'$
GZ1, GZ2	$= G_1$
GZP1, GZP2	$= \partial G_1/\partial z'$
GZZ1, GZZ2	$= \partial G_0/\partial z'$
IJ	= IJX = 0 to indicate that the field point is on the source segment
INX1, INX2	= 2 to use the thin wire form at end 1 or end 2, respectively
IRA	= 1 to indicate RHX < BX
RH	$= \rho$ coordinate of the field point or wire radius
RHK	= k(RH)
RHX	$= \rho$ coordinate of the field point
RKB2	$= (\mathrm{RHK})^2$
S	$= \Delta$
SH	$= \Delta/2$
SHK	$= k\Delta/2$
SINT	$= \int_{-\Delta/2}^{\Delta/2} \sin(kr)/r \, dz$
SS	$= \sin(k\Delta/2)$
XK	$= k = 2\pi/\lambda,\ \lambda = 1$
Z	= z coordinate of field point
Z1	$= -\Delta/2 - z$
Z2	$= \Delta/2 - z$
ZPK	$= kz$

CONSTANT

$4.77134118 = \eta/(8\pi^2)$

```
      SUBROUTINE EKSCX (BX,S,Z,RHX,XK,IJ,INX1,INX2,EZS,ERS,EZC,ERC,EZK,E
     1RK)
C     COMPUTE E FIELD OF SINE, COSINE, AND CONSTANT CURRENT FILAMENTS BY
C     EXTENDED THIN WIRE APPROXIMATION.
      COMPLEX CON,GZ1,GZ2,GZP1,GZP2,GR1,GR2,GRP1,GRP2,EZS,EZC,ERS,ERC,GR
     1K1,GRK2,EZK,ERK,GZZ1,GZZ2
      COMMON /TMI/ ZPK,RKB2,IJX
      DIMENSION CONX(2)
      EQUIVALENCE (CONX,CON)
      DATA CONX/0.,4.771341189/
      IF (RHX.LT.BX) GO TO 1
      RH=RHX
      B=BX
      IRA=0
      GO TO 2
    1 RH=BX
      B=RHX
      IRA=1
    2 SH=.5*S
      IJX=IJ
      ZPK=XK*Z
      RHK=XK*RH
      RKB2=RHK*RHK
      SHK=XK*SH
      SS=SIN(SHK)
      CS=COS(SHK)
      Z2=SH-Z
      Z1=-(SH+Z)
      A2=B*B
      IF (INX1.EQ.2) GO TO 3
      CALL GXX (Z1,RH,B,A2,XK,IRA,GZ1,GZP1,GR1,GRP1,GRK1,GZZ1)
      GO TO 4
    3 CALL GX (Z1,RHX,XK,GZ1,GRK1)
      GZP1=GRK1*Z1
      GR1=GZ1/RHX
      GRP1=GZP1/RHX
      GRK1=GRK1*RHX
      GZZ1=(0.,0.)
    4 IF (INX2.EQ.2) GO TO 5
      CALL GXX (Z2,RH,B,A2,XK,IRA,GZ2,GZP2,GR2,GRP2,GRK2,GZZ2)
      GO TO 6
    5 CALL GX (Z2,RHX,XK,GZ2,GRK2)
      GZP2=GRK2*Z2
      GR2=GZ2/RHX
      GRP2=GZP2/RHX
      GRK2=GRK2*RHX
      GZZ2=(0.,0.)
    6 EZS=CON*((GZ2-GZ1)*CS*XK-(GZP2+GZP1)*SS)
      EZC=-CON*((GZ2+GZ1)*SS*XK+(GZP2-GZP1)*CS)
      ERS=-CON*((Z2*GRP2+Z1*GRP1+GR2+GR1)*SS-(Z2*GR2-Z1*GR1)*CS*XK)
      ERC=-CON*((Z2*GRP2-Z1*GRP1+GR2-GR1)*CS+(Z2*GR2+Z1*GR1)*SS*XK)
      ERK=CON*(GRK2-GRK1)
      CALL INTX (-SHK,SHK,RHK,IJ,CINT,SINT)
      BK=B*XK
      BK2=BK*BK*.25
      EZK=-CON*(GZP2-GZP1+XK*XK*(1.-BK2)*CMPLX(CINT,-SINT)-BK2*(GZZ2-GZZ
     11))
      RETURN
      END
```

ENF

PURPOSE

To check for an end of file.

METHOD

ENF uses the standard Fortran end-of-file test and returns the logical values .TRUE. or .FALSE. This separate function is used for convenience in adapting the code to particular computers, since the Fortran end-of-file test statements often differ between computers. The form of ENF here is for CDC computers.

SYMBOL DICTIONARY

ENF = logical value: .TRUE. if end of file was encountered; .FALSE. otherwise

NUNIT = logical unit number

CODE LISTING

```
      LOGICAL FUNCTION ENF(NUNIT)                                EN   1
      IF (EOF,NUNIT) 1,2                                         EN   2
    1 ENF=.TRUE.                                                 EN   3
      RETURN                                                     EN   4
    2 ENF=.FALSE.                                                EN   5
      RETURN                                                     EN   6
      END                                                        EN   7-
```

ETMNS

PURPOSE

To fill the array representing the right-hand side of the matrix equation with the negative of the electric field tangent to the segments and with the tangential magnetic field on the surfaces.

METHOD

The array E represents the right-hand side of the matrix equation. For the i^{th} segment, the right-hand side is the negative of the applied electric field component tangent to the segment, and is stored in location i in array E. For the i^{th} surface patch, there are two rows in the matrix equation (from the two components of the vector equations) with locations N + 2i - 1 and N + 2i, where N is the total number of wire segments. The contents of E for these locations are

$$E\ (N + 2i - 1) = -\hat{t}_1 \cdot (\hat{n} \times \overline{H}_i) = \pm t_2 \cdot \overline{H}_i$$

$$E\ (N + 2i) = \hat{t}_2 \cdot (\hat{n} \times \overline{H}_i) = \pm t_1 \cdot \overline{H}_i$$

where $\overline{H}_i$ is the magnetic field applied to patch i. The forms on the right are used in the code with the plus sign applying when $(\hat{t}_1, \hat{t}_2, \hat{n})$ forms a right-hand system and the minus sign when left-hand. To avoid the need to check $(\hat{t}_1, \hat{t}_2, \hat{n})$, the sign is stored in array SALP where, for patch i, SALP (LD + 1 - i) = ±1 according to $(\hat{t}_1, \hat{t}_2, \hat{n})$, with LD the length of the arrays in COMMON/DATA/. If the structure has symmetry, the entries in E are reordered by subroutine SOLVES.

The parameter IPR selects the type of excitation; the meanings of other parameters depend on the option selected by IPR and are explained below. The excitations associated with IPR values are:

IPR = 0 applied field voltage source
- 1 incident plane wave, linear polarization
- 2 incident plane wave, right-hand elliptic polarization
- 3 incident plane wave, left-hand elliptic polarization
- 4 infinitesimal current element source
- 5 current slope discontinuity voltage source

CODING

ET29 - ET34 Applied field voltage source (IPR = 0).

ET36 - ET38 QDSRC is called for each current slope discontinuity voltage source (IPR = 5).

ET44 - ET160 Incident plane wave. The direction of propagation and polarization of the wave are illustrated in figure 4 in which $\hat{p}$ is the unit vector normal to $\hat{k}$ in the plane defined by $\hat{k}$ and $\hat{z}$. The plane wave as a function of position $\overline{r}$ is

$$\overline{E}^I(\overline{r}) = \overline{E}_0 \exp(-j\overline{k}\cdot\overline{r})$$

$$\overline{H}^I(\overline{r}) = \frac{1}{\eta}\hat{k} \times \overline{E}_0 \exp(-j\overline{k}\cdot\overline{r})$$

where

$\overline{k} = (2\pi/\lambda)\,\hat{k}$

$\hat{k}$ = unit vector in direction of propagation

$\overline{E}_0 = \hat{E}_1$ for linear polarization

$= (\hat{E}_1 - jA\hat{E}_2)$ for right-hand elliptical polarization

$= (\hat{E}_1 + jA\hat{E}_2)$ for left-hand elliptical polarization

A = ellipse axes ratio

$\hat{E}_2 = \hat{k} \times \hat{E}_1$

ET44 - ET58 P1 = θ

P2 = ϕ

P3 = ξ

PX, PY, PZ = x, y, z components of $\hat{E}_1$

WX, WY, WZ = $\hat{k}$

QX, QY, QZ = $\hat{E}_2 = \hat{k} \times \hat{E}_1$

ET61 - ET68 Ground reflection coefficients computed:

RRH = reflection coefficient for E normal to the plane of incidence

RRV = reflection coefficient for E in the plane of incidence

ET70 - ET108 Linearly polarized wave (IPR = 1).

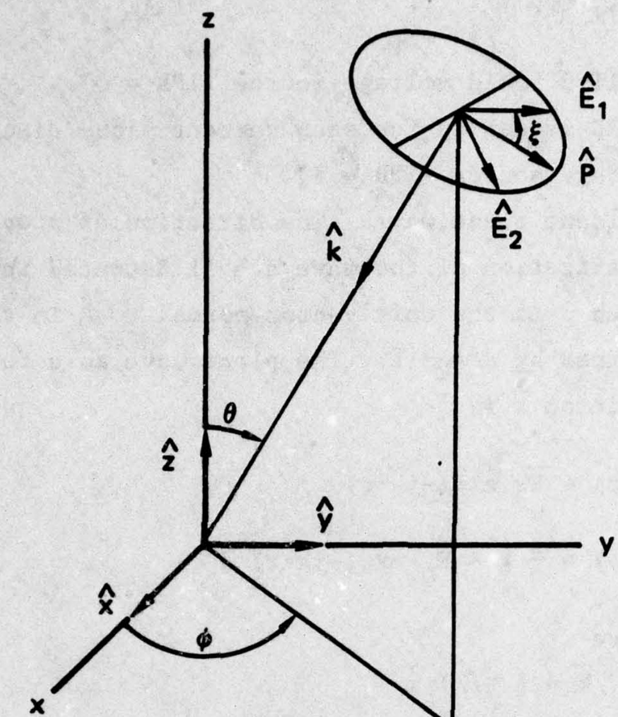


Figure 4. Coordinate Parameters for the Incident Plane Wave.

ET71 - ET73	Direct illumination of segments by E field. ARG = $-\bar{k} \cdot \bar{r}_i$, where $\bar{r}_i$ = center point of segment I. E(I) = $-(\hat{E}_1 \cdot \hat{i}) \exp(-j\bar{k} \cdot \bar{r}_i)$, where $\hat{i}$ = unit vector in the direction of segment I.
ET75 - ET82	Illumination of segments by the ground reflected field. CX, CY, CZ = reflected E field
ET84 - ET93	Direct H field illumination of patches.
ET95 - ET108	Illumination of patches by the ground reflected field. CX, CY, CZ = reflected H field
ET113 - ET159	Elliptically polarized wave (IPR = 2 or 3). P6 = ellipse axes ratio = A.
ET116 - ET121	Direct E field illumination of segments. CX, CY, CZ = $\hat{E}_1 \pm jA\hat{E}_2$ (+ for left-hand polarization, - for right-hand)
ET123 - ET130	Illumination of segments by the ground reflected E field.
ET132 - ET144	Illumination of patches by the direct H field. CX, CY, CZ = $\hat{k} \times \bar{E}_0$
ET146 - ET159	Illumination of patches by ground reflected H field.

ET164 - ET225 Infinitesimal current element source (IPR = 4). A current element of moment $I_0 \ell$ at the origin of a spherical coordinate system, as shown in figure 5, produces field components

$$\overline{E}_R(\overline{R}) = I_0 \ell \frac{\eta}{2\pi} \exp(-jkR) \left(1 - \frac{j}{kR}\right) \frac{1}{R^2} \cos\theta \, \hat{R}$$

$$\overline{E}_\theta(\overline{R}) = I_0 \ell \frac{\eta}{4\pi} \exp(-jkR) \left[\frac{jk}{R} + \left(1 - \frac{j}{kR}\right) \frac{1}{R^2}\right] \sin\theta \, \hat{\theta}$$

$$H_\phi = \frac{I_0 \ell}{4\pi} \exp(-jkR) \left(\frac{1}{R^2} + \frac{jk}{R}\right) \sin\theta$$

If the location and orientation of segment i and the current element with respect to the x, y, z coordinate system are

$\overline{r}_i$ = location of segment i
$\hat{i}$ = orientation of segment i
$\overline{D}$ = location of current element
$\hat{d}$ = orientation of current element

then

$\overline{R} = \overline{r}_i - \overline{D}$
$\hat{R} = \overline{R} / |\overline{R}|$
$\cos\theta = \hat{R} \cdot \hat{d}$
$\sin\theta = [1 - \cos^2\theta]^{1/2}$

The orientation of the current element is defined by its angle of elevation above the x-y plane, a, and the angle from the x axis to its projection on the x-y plane, b. Thus, $\hat{d} = \cos a \cos b \, \hat{x} + \cos a \sin b \, \hat{y} + \sin a \, \hat{z}$. The $\hat{R}$ and $\hat{\theta}$ field components are converted to $\hat{\rho}$ and $\hat{d}$ components E_ρ and E_d, where

$$E_d = E_R \cos\theta - E_\theta \sin\theta$$

$$E_\rho = E_R \sin\theta + E_\theta \cos\theta$$

and the excitation computed as

$$E(I) = -\hat{i} \cdot (E_d \hat{d} + E_\rho \hat{\rho}) .$$

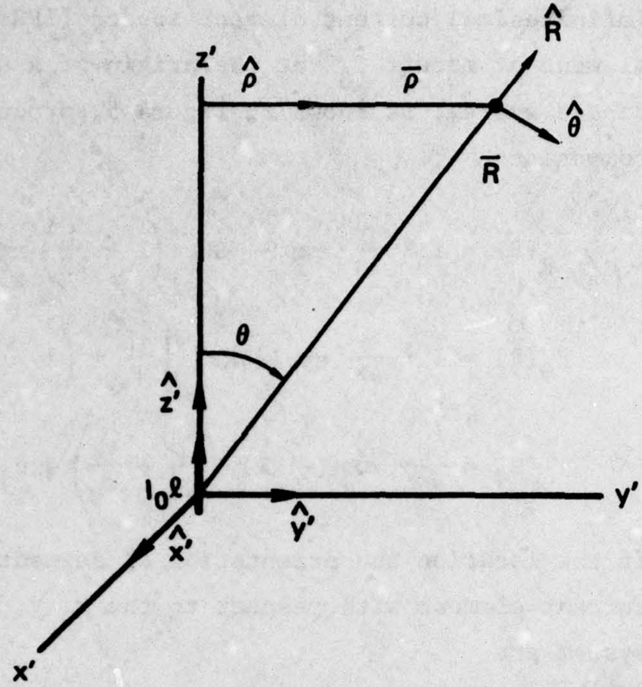


Figure 5. Coordinate Parameters for Current Element.

ET164 - ET225 P1, P2, P3 = x, y, z coordinates of current element ($\overline{D}$)

P4 = a

P5 = b

P6 = $I_0\ell/\lambda^2$

ET164 - ET169 WX, WY, WZ = x, y, and z components of $\hat{d}$

DS = $(\eta/2\pi)\ I_0\ell/\lambda^2$

DSH = $(1/4\pi)\ I_0\ell/\lambda^2$

ET173 Start of loop over all segments and patches.

ET176 - ET179 For patches,

IS = location of patch data in geometry arrays

I1, I2 = locations to be filled in E

ET180 - ET182 PX, PY, PZ = $\overline{R}/\lambda$

ET183 - ET193 R = $|\overline{R}/\lambda|$

PX, PY, PZ = $\hat{R}$

CTH = cos θ

STH = sin θ

QX, QY, QZ = $\hat{R} - (\hat{d} \cdot \hat{R})\hat{d}$

ET196 - ET204 QX, QY, QZ = $\hat{\rho}$

T1 = exp(-jk R)

ET206 - ET215 E field on segments

T2 = $(1 - j/kR)\lambda^2/R^2$

ER = E_R

ET = E_θ

ERH = E_ρ

EZH = E_z

CX, CY, CZ = x, y, z components of total E field

ET216 - ET224 H field on patches

PX, PY, PZ = $\hat{d} \times \hat{\rho} = \hat{\phi}$

T2 = $\pm H_\phi$

CX, CY, CZ = $\pm \bar{H}^I$

CONSTANTS

1.E-30 = tolerance in test for zero

2.654420938E-3 = $1/\eta = \sqrt{\epsilon_0/\mu_0}$

59.958 = $\eta/2\pi$

6.283185308 = 2π

```
      SUBROUTINE ETMNS (P1,P2,P3,P4,P5,P6,IPR,E)                        ET   1
C                                                                       ET   2
C     ETMNS FILLS THE ARRAY E WITH THE NEGATIVE OF THE ELECTRIC FIELD   ET   3
C     INCIDENT ON THE STRUCTURE.  E IS THE RIGHT HAND SIDE OF THE MATRIX ET   4
C     EQUATION.                                                         ET   5
C                                                                       ET   6
      COMPLEX E,CX,CY,CZ,VSANT,TX1,TX2,ER,ET,EZH,ERH,VQD,VQDS,ZRATI,ZRAT ET   7
     1I2,RRV,RRH,T1,TT1,TT2,FRATI                                       ET   8
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 ET   9
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( ET  10
     2300),WLAM,IPSYM                                                   ET  11
      COMMON /ANGL/ SALP(300)                                           ET  12
      COMMON /VSORC/ VQD(30),VSANT(30),VQDS(30),IVQD(30),ISANT(30),IQDS( ET  13
     130),NVQD,NSANT,NQDS                                               ET  14
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR, ET  15
     1IPERF,T1,T2                                                       ET  16
      DIMENSION CAB(1), SAB(1), E(600)                                  ET  17
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)          ET  18
      EQUIVALENCE (CAB,ALP), (SAB,BET)                                  ET  19
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON ET  20
     12), (T2Z,ITAG)                                                    ET  21
      DATA TP/6.283185308/,RETA/2.654420938E-3/                         ET  22
      NEQ=N+2*M                                                         ET  23
      NQDS=0                                                            ET  24
      IF (IPR.GT.0.AND.IPR.NE.5) GO TO 5                                ET  25
C                                                                       ET  26
C     APPLIED FIELD OF VOLTAGE SOURCES FOR TRANSMITTING CASE            ET  27
C                                                                       ET  28
      DO 1 I=1,NEQ                                                      ET  29
1     E(I)=(0.,0.)                                                      ET  30
      IF (NSANT.EQ.0) GO TO 3                                           ET  31
      DO 2 I=1,NSANT                                                    ET  32
      IS=ISANT(I)                                                       ET  33
2     E(IS)=-VSANT(I)/(SI(IS)*WLAM)                                     ET  34
3     IF (NVQD.EQ.0) RETURN                                             ET  35
      DO 4 I=1,NVQD                                                     ET  36
      IS=IVQD(I)                                                        ET  37
4     CALL QDSRC (IS,VQD(I),E)                                          ET  38
      RETURN                                                            ET  39
5     IF (IPR.GT.3) GO TO 19                                            ET  40
C                                                                       ET  41
C     INCIDENT PLANE WAVE, LINEARLY POLARIZED.                          ET  42
C                                                                       ET  43
      CTH=COS(P1)                                                       ET  44
      STH=SIN(P1)                                                       ET  45
      CPH=COS(P2)                                                       ET  46
      SPH=SIN(P2)                                                       ET  47
      CET=COS(P3)                                                       ET  48
      SET=SIN(P3)                                                       ET  49
      PX=CTH*CPH*CET-SPH*SET                                            ET  50
      PY=CTH*SPH*CET+CPH*SET                                            ET  51
      PZ=-STH*CET                                                       ET  52
      WX=-STH*CPH                                                       ET  53
      WY=-STH*SPH                                                       ET  54
      WZ=-CTH                                                           ET  55
      QX=WY*PZ-WZ*PY                                                    ET  56
      QY=WZ*PX-WX*PZ                                                    ET  57
      QZ=WX*PY-WY*PX                                                    ET  58
      IF (KSYMP.EQ.1) GO TO 7                                           ET  59
      IF (IPERF.EQ.1) GO TO 6                                           ET  60
      RRV=CSQRT(1.-ZRATI*ZRATI*STH*STH)                                 ET  61
      RRH=ZRATI*CTH                                                     ET  62
      RRH=(RRH-RRV)/(RRH+RRV)                                           ET  63
      RRV=ZRATI*RRV                                                     ET  64
```

```
      RRV=-(CTH-RRV)/(CTH+RRV)
      GO TO 7
6     RRV=-(1.,0.)
      RRH=-(1.,0.)
7     IF (IPR.GT.1) GO TO 13
      IF (N.EQ.0) GO TO 10
      DO 8 I=1,N
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
8     E(I)=-(PX*CAB(I)+PY*SAB(I)+PZ*SALP(I))*CMPLX(COS(ARG),SIN(ARG))
      IF (KSYMP.EQ.1) GO TO 10
      TT1=(PY*CPH-PX*SPH)*(RRH-RRV)
      CX=RRV*PX-TT1*SPH
      CY=RRV*PY+TT1*CPH
      CZ=-RRV*PZ
      DO 9 I=1,N
      ARG=-TP*(WX*X(I)+WY*Y(I)-WZ*Z(I))
9     E(I)=E(I)-(CX*CAB(I)+CY*SAB(I)+CZ*SALP(I))*CMPLX(COS(ARG),SIN(ARG)
     1)
10    IF (M.EQ.0) RETURN
      I=LD+1
      I1=N-1
      DO 11 IS=1,M
      I=I-1
      I1=I1+2
      I2=I1+1
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
      TT1=CMPLX(COS(ARG),SIN(ARG))*SALP(I)*RETA
      E(I2)=(QX*T1X(I)+QY*T1Y(I)+QZ*T1Z(I))*TT1
11    E(I1)=(QX*T2X(I)+QY*T2Y(I)+QZ*T2Z(I))*TT1
      IF (KSYMP.EQ.1) RETURN
      TT1=(QY*CPH-QX*SPH)*(RRV-RRH)
      CX=-(RRH*QX-TT1*SPH)
      CY=-(RRH*QY+TT1*CPH)
      CZ=RRH*QZ
      I=LD+1
      I1=N-1
      DO 12 IS=1,M
      I=I-1
      I1=I1+2
      I2=I1+1
      ARG=-TP*(WX*X(I)+WY*Y(I)-WZ*Z(I))
      TT1=CMPLX(COS(ARG),SIN(ARG))*SALP(I)*RETA
      E(I2)=E(I2)+(CX*T1X(I)+CY*T1Y(I)+CZ*T1Z(I))*TT1
12    E(I1)=E(I1)+(CX*T2X(I)+CY*T2Y(I)+CZ*T2Z(I))*TT1
      RETURN
C
C     INCIDENT PLANE WAVE, ELLIPTIC POLARIZATION.
C
13    TT1=-(0.,1.)*P6
      IF (IPR.EQ.3) TT1=-TT1
      IF (N.EQ.0) GO TO 16
      CX=PX+TT1*QX
      CY=PY+TT1*QY
      CZ=PZ+TT1*QZ
      DO 14 I=1,N
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
14    E(I)=-(CX*CAB(I)+CY*SAB(I)+CZ*SALP(I))*CMPLX(COS(ARG),SIN(ARG))
      IF (KSYMP.EQ.1) GO TO 16
      TT2=(CY*CPH-CX*SPH)*(RRH-RRV)
      CX=RRV*CX-TT2*SPH
      CY=RRV*CY+TT2*CPH
      CZ=-RRV*CZ
      DO 15 I=1,N
      ARG=-TP*(WX*X(I)+WY*Y(I)-WZ*Z(I))
```

```
15    E(I)=E(I)-(CX*CAB(I)+CY*SAB(I)+CZ*SALP(I))*CMPLX(COS(ARG),SIN(ARG)
     1)
16    IF (M.EQ.0) RETURN
      CX=QX-TT1*PX
      CY=QY-TT1*PY
      CZ=QZ-TT1*PZ
      I=LD+1
      I1=N-1
      DO 17 IS=1,M
      I=I-1
      I1=I1+2
      I2=I1+1
      ARG=-TP*(WX*X(I)+WY*Y(I)+WZ*Z(I))
      TT2=CMPLX(COS(ARG),SIN(ARG))*SALP(I)*RETA
      E(I2)=(CX*T1X(I)+CY*T1Y(I)+CZ*T1Z(I))*TT2
17    E(I1)=(CX*T2X(I)+CY*T2Y(I)+CZ*T2Z(I))*TT2
      IF (KSYMP.EQ.1) RETURN
      TT1=(CY*CPH-CX*SPH)*(RRV-RRH)
      CX=-(RRH*CX-TT1*SPH)
      CY=-(RRH*CY+TT1*CPH)
      CZ=RRH*CZ
      I=LD+1
      I1=N-1
      DO 18 IS=1,M
      I=I-1
      I1=I1+2
      I2=I1+1
      ARG=-TP*(WX*X(I)+WY*Y(I)-WZ*Z(I))
      TT1=CMPLX(COS(ARG),SIN(ARG))*SALP(I)*RETA
      E(I2)=E(I2)+(CX*T1X(I)+CY*T1Y(I)+CZ*T1Z(I))*TT1
18    E(I1)=E(I1)+(CX*T2X(I)+CY*T2Y(I)+CZ*T2Z(I))*TT1
      RETURN
C
C     INCIDENT FIELD OF AN ELEMENTARY CURRENT SOURCE.
C
19    WZ=COS(P4)
      WX=WZ*COS(P5)
      WY=WZ*SIN(P5)
      WZ=SIN(P4)
      DS=P6*59.958
      DSH=P6/(2.*TP)
      NPM=N+M
      IS=LD+1
      I1=N-1
      DO 24 I=1,NPM
      II=I
      IF (I.LE.N) GO TO 20
      IS=IS-1
      II=IS
      I1=I1+2
      I2=I1+1
20    PX=X(II)-P1
      PY=Y(II)-P2
      PZ=Z(II)-P3
      RS=PX*PX+PY*PY+PZ*PZ
      IF (RS.LT.1.E-30) GO TO 24
      R=SQRT(RS)
      PX=PX/R
      PY=PY/R
      PZ=PZ/R
      CTH=PX*WX+PY*WY+PZ*WZ
      STH=SQRT(1.-CTH*CTH)
      QX=PX-WX*CTH
      QY=PY-WY*CTH
```

```
      QZ=PZ-WZ*CTH
      ARG=SQRT(QX*QX+QY*QY+QZ*QZ)
      IF (ARG.LT.1.E-30) GO TO 21
      QX=QX/ARG
      QY=QY/ARG
      QZ=QZ/ARG
      GO TO 22
21    QX=1.
      QY=0.
      QZ=0.
22    ARG=-TP*R
      TT1=CMPLX(COS(ARG),SIN(ARG))
      IF (I.GT.N) GO TO 23
      TT2=CMPLX(1.,-1./(R*TP))/RS
      ER=DS*TT1*TT2*CTH
      ET=.5*DS*TT1*((0.,1.)*TP/R+TT2)*STH
      EZH=ER*CTH-ET*STH
      ERH=ER*STH+ET*CTH
      CX=EZH*WX+ERH*QX
      CY=EZH*WY+ERH*QY
      CZ=EZH*WZ+ERH*QZ
      E(I)=-(CX*CAB(I)+CY*SAB(I)+CZ*SALP(I))
      GO TO 24
23    PX=WY*QZ-WZ*QY
      PY=WZ*QX-WX*QZ
      PZ=WX*QY-WY*QX
      TT2=DSH*TT1*CMPLX(1./R,TP)/R*STH*SALP(II)
      CX=TT2*PX
      CY=TT2*PY
      CZ=TT2*PZ
      E(I2)=CX*T1X(II)+CY*T1Y(II)+CZ*T1Z(II)
      E(I1)=CX*T2X(II)+CY*T2Y(II)+CZ*T2Z(II)
24    CONTINUE
      RETURN
      END
```

FACGF

PURPOSE

To perform the steps in the NGF solution that do not depend on the excitation vector.

METHOD

The NGF solution procedure is discussed in Section VI. The steps performed in FACGF are to evaluate $A^{-1}B$ and $D - CA^{-1}B$. The matrix $D - CA^{-1}B$ is then factored into triangular matrices L and U. The procedure is complicated by the possible need to use file storage for the matrices. The comments in the code and the tables for ICASX = 2, 3 and 4 in Section VII offer a fairly complete description of the procedure.

SYMBOL DICTIONARY

A	= array for matrix A (L U factors) or block of A if file storage is used
B	= array for B or block of B
BX	= array for B when $A^{-1}B$ is being computed with ICASX = 2. The array B starts at the beginning of CM in this case. BX leaves room for A_F at the beginning of CM
C	= array for C or block of C (matrix transposed)
D	= array for D or block of D (matrix transposed)
IBFL	= file on which B is stored
ICASS	= saved value of ICASE
IP	= pivot index array
IX	= data on row interchanges in LFACTR
M1	= number of patches in the NGF
MP	= number of patches in a symmetric section in the NGF
N1	= number of segments in the NGF
N1C	= number of columns in C (same as order of A)
N1CP	= N1C + 1
N2C	= order of matrix D
NBLSYS	= saved value of NBLSYM
NIC	= index increment
NLSYS	= saved value of NLSYM

NP = number of segments in a symmetric section in the NGF

NPSYS = saved value of NPSYM

SUM = summation variable for matrix products

```
      SUBROUTINE FACGF (A,B,C,D,BX,IP,IX,NP,N1,MP,M1,N1C,N2C)           FG   1
C     FACGF COMPUTES AND FACTORS D-C(INV(A)B).                          FG   2
      COMPLEX A,B,C,D,BX,SUM                                            FG   3
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I FG   4
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL                                FG   5
      DIMENSION A(1), B(N1C,1), C(N1C,1), D(N2C,1), BX(N1C,1), IP(1), IX FG   6
     1(1)                                                               FG   7
      IF (N2C.EQ.0) RETURN                                              FG   8
      IBFL=14                                                           FG   9
      IF (ICASX.LT.3) GO TO 1                                           FG  10
C     CONVERT B FROM BLOCKS OF ROWS ON T14 TO BLOCKS OF COL. ON T16     FG  11
      CALL REBLK (B,C,N1C,NPBX,N2C)                                     FG  12
      IBFL=16                                                           FG  13
1     NPB=NPBL                                                          FG  14
      IF (ICASX.EQ.2) REWIND 14                                         FG  15
C     COMPUTE INV(A)B AND WRITE ON TAPE14                               FG  16
      DO 2 IB=1,NBBL                                                    FG  17
      IF (IB.EQ.NBBL) NPB=NLBL                                          FG  18
      IF (ICASX.GT.1) READ (IBFL) ((BX(I,J),I=1,N1C),J=1,NPB)           FG  19
      CALL SOLVES (A,IP,BX,N1C,NPB,NP,N1,MP,M1,13,13)                   FG  20
      IF (ICASX.EQ.2) REWIND 14                                         FG  21
      IF (ICASX.GT.1) WRITE (14) ((BX(I,J),I=1,N1C),J=1,NPB)            FG  22
2     CONTINUE                                                          FG  23
      IF (ICASX.EQ.1) GO TO 3                                           FG  24
      REWIND 11                                                         FG  25
      REWIND 12                                                         FG  26
      REWIND 15                                                         FG  27
      REWIND IBFL                                                       FG  28
3     NPC=NPBL                                                          FG  29
C     COMPUTE D-C(INV(A)B) AND WRITE ON TAPE11                          FG  30
      DO 8 IC=1,NBBL                                                    FG  31
      IF (IC.EQ.NBBL) NPC=NLBL                                          FG  32
      IF (ICASX.EQ.1) GO TO 4                                           FG  33
      READ (15) ((C(I,J),I=1,N1C),J=1,NPC)                              FG  34
      READ (12) ((D(I,J),I=1,N2C),J=1,NPC)                              FG  35
      REWIND 14                                                         FG  36
4     NPB=NPBL                                                          FG  37
      NIC=0                                                             FG  38
      DO 7 IB=1,NBBL                                                    FG  39
      IF (IB.EQ.NBBL) NPB=NLBL                                          FG  40
      IF (ICASX.GT.1) READ (14) ((B(I,J),I=1,N1C),J=1,NPB)              FG  41
      DO 6 I=1,NPB                                                      FG  42
      II=I+NIC                                                          FG  43
      DO 6 J=1,NPC                                                      FG  44
      SUM=(0.,0.)                                                       FG  45
      DO 5 K=1,N1C                                                      FG  46
5     SUM=SUM+B(K,I)*C(K,J)                                             FG  47
6     D(II,J)=D(II,J)-SUM                                               FG  48
7     NIC=NIC+NPBL                                                      FG  49
      IF (ICASX.GT.1) WRITE (11) ((D(I,J),I=1,N2C),J=1,NPBL)            FG  50
8     CONTINUE                                                          FG  51
      IF (ICASX.EQ.1) GO TO 9                                           FG  52
      REWIND 11                                                         FG  53
      REWIND 12                                                         FG  54
      REWIND 14                                                         FG  55
      REWIND 15                                                         FG  56
9     N1CP=N1C+1                                                        FG  57
C     FACTOR D-C(INV(A)B)                                               FG  58
      IF (ICASX.GT.1) GO TO 10                                          FG  59
      CALL FACTR (N2C,D,IP(N1CP),N2C)                                   FG  60
      GO TO 13                                                          FG  61
10    IF (ICASX.EQ.4) GO TO 12                                          FG  62
      NPB=NPBL                                                          FG  63
      IC=0                                                              FG  64
```

```
      DO 11 IB=1,NBBL                                                   FG  65
      IF (IB.EQ.NBBL) NPB=NLBL                                          FG  66
      II=IC+1                                                           FG  67
      IC=IC+N2C*NPB                                                     FG  68
   11 READ (11) (B(I,1),I=II,IC)                                        FG  69
      REWIND 11                                                         FG  70
      CALL FACTR (N2C,B,IP(N1CP),N2C)                                   FG  71
      NIC=N2C*N2C                                                       FG  72
      WRITE (11) (B(I,1),I=1,NIC)                                       FG  73
      REWIND 11                                                         FG  74
      GO TO 13                                                          FG  75
   12 NBLSYS=NBLSYM                                                     FG  76
      NPSYS=NPSYM                                                       FG  77
      NLSYS=NLSYM                                                       FG  78
      ICASS=ICASE                                                       FG  79
      NBLSYM=NBBL                                                       FG  80
      NPSYM=NPBL                                                        FG  81
      NLSYM=NLBL                                                        FG  82
      ICASE=3                                                           FG  83
      CALL FACIO (B,N2C,1,IX(N1CP),11,12,16,11)                         FG  84
      CALL LUNSCR (B,N2C,1,IP(N1CP),IX(N1CP),12,11,16)                  FG  85
      NBLSYM=NBLSYS                                                     FG  86
      NPSYM=NPSYS                                                       FG  87
      NLSYM=NLSYS                                                       FG  88
      ICASE=ICASS                                                       FG  89
   13 RETURN                                                            FG  90
      END                                                               FG  91-
```

FACIO

PURPOSE

To read and write matrix blocks needed for the LU decomposition.

METHOD

Sequential access is used on all files. The matrix is initially stored on file IU1 in blocks of columns of the transposed matrix. The block size is such that two blocks will fit into the array A for the Gauss elimination process. If the matrix were divided into four blocks, the order for reading the blocks into core would be

Blocks	
1, 2	1 and 2 will be completely factored
1, 3	3 and 4 partially factored
1, 4	
2, 3	factorization of 3 completed
2, 4	4 partially factored
3, 4	factorization complete

IU1 is the initial input file. Partially factored blocks are read from file IFILE3 and written to IFILE4 where IFILE3 = IU3 and IFILE4 = IU4 when IXBLK1 is odd, and IFILE3 = IU4 and IFILE4 = IU3 when IXBLK1 is even. Completed blocks are written to file IU2. Although the last block may be shorter than other blocks the same number of words is read or written. The excess words are ignored in subroutine LFACTR.

Subroutine LFACTR is called to perform the Gauss elimination. For a symmetric structure the loop from F018 to F043 factors each submatrix.

SYMBOL DICTIONARY

A	= array for matrix storage
I1	= location in A of beginning of block 1
I2	= location in A of end of block 1
I3	= location in A of beginning of block 2
I4	= location in A of end of block 2
IFILE3	= input file
IFILE4	= output file
IP	= array for pivot element indices

IT	= number of words in a matrix block
IU1, IU2, IU3, IU4	= file numbers
IXBLK1	= number of first block stored in A
IXBLK2	= number of second block stored in A
KA	= first location in IP for submatrix KK
NBM	= number of blocks minus one
NOP	= number of submatrices for symmetry
NROW	= number of rows in a block
T1, T2, TIME	= variables to sum total time spent in LFACTR

```
      SUBROUTINE FACIO (A,NROW,NOP,IP,IU1,IU2,IU3,IU4)
C
C     FACIO CONTROLS I/O FOR OUT-OF-CORE FACTORIZATION
C
      COMPLEX A
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      DIMENSION A(NROW,1), IP(NROW)
      IT=2*NPSYM*NROW
      NBM=NBLSYM-1
      I1=1
      I2=IT
      I3=I2+1
      I4=2*IT
      TIME=0.
      REWIND IU1
      REWIND IU2
      DO 3 KK=1,NOP
      KA=(KK-1)*NROW+1
      IFILE3=IU1
      IFILE4=IU3
      DO 2 IXBLK1=1,NBM
      REWIND IU3
      REWIND IU4
      CALL BLCKIN (A,IFILE3,I1,I2,1,17)
      IXBP=IXBLK1+1
      DO 1 IXBLK2=IXBP,NBLSYM
      CALL BLCKIN (A,IFILE3,I3,I4,1,18)
      CALL SECOND (T1)
      CALL LFACTR (A,NROW,IXBLK1,IXBLK2,IP(KA))
      CALL SECOND (T2)
      TIME=TIME+T2-T1
      IF (IXBLK2.EQ.IXBP) CALL BLCKOT (A,IU2,I1,I2,1,19)
      IF (IXBLK1.EQ.NBM.AND.IXBLK2.EQ.NBLSYM) IFILE4=IU2
      CALL BLCKOT (A,IFILE4,I3,I4,1,20)
    1 CONTINUE
      IFILE3=IU3
      IFILE4=IU4
      IF ((IXBLK1/2)*2.NE.IXBLK1) GO TO 2
      IFILE3=IU4
      IFILE4=IU3
    2 CONTINUE
    3 CONTINUE
      REWIND IU1
      REWIND IU2
      REWIND IU3
      REWIND IU4
      PRINT 4, TIME
      RETURN
C
    4 FORMAT (35H CP TIME TAKEN FOR FACTORIZATION = ,E12.5)
      END
```

FACTR

PURPOSE

To factor a complex matrix into a lower triangular and an upper triangular matrix using the Gauss-Doolittle technique. The matrix in this case is a transposed matrix. The factored matrix is used by subroutine SOLVE to determine the solution of the matrix equation Ax = B.

METHOD

The algorithm used in this routine is presented by A. Ralston (ref. 1). The decomposition of the matrix A is such that A = LU, where L is a lower triangular matrix with 1's down the diagonal, and U is an upper triangular matrix. The L and U matrices overwrite the matrix A. The computations to obtain L and U are done using one complex scratch vector (D) and one integer vector (IP) that keep track of row interchanges when elements are positioned for size. If positioning for size is not taken into account, the general procedure is

$$a_{11} = u_{11}$$

$$a_{i1} = \ell_{i1} u_{11} \qquad i = 2, \ldots, n$$

which gives the first column of the L and U matrices. Then

$$a_{12} = u_{12}$$

$$a_{22} = \ell_{21} u_{12} + u_{22}$$

$$a_{i2} = \ell_{i1} u_{12} + \ell_{i2} u_{22} \qquad i = 3, \ldots, n$$

gives the second column. The computations for the successive columns continue in this way. The general equations for the r^{th} column are

$$a_{1r} = u_{1r}$$

$$a_{2r} = \ell_{21} u_{1r} + u_{2r}$$

$$\vdots$$

$$a_{rr} = \ell_{r1} u_{1r} + \ell_{r2} u_{2r} + \cdots + \ell_{r,r-1} u_{r-1,r} + u_{rr}$$

$$a_{ir} = \ell_{i1} u_{1r} + \cdots + \ell_{ir} u_{rr}, \qquad i = r + 1, \ldots, n$$

FACTR

There are only two differences in the coding used in FACTR and the coding suggested by Ralston. The first is that double precision variables are not used for the accumulation of sums, since for the size and conditioning of the matrices anticipated in core, the computer word length is sufficient to insure accuracy. The second difference is that the row and column indices of the A matrix in the routine have been interchanged to handle the transposed matrix.

CODING

The coding is divided into five steps which correspond to the steps given by Ralston.

FA14 Loop over columns (rows with the interchanged indices used in the routine).

FA18 - FA20 Fill D vector with column (row) of A.

FA24 - FA35 Solution for u_{ir} $(i = 1,\ldots,r)$ in the above equations taking into account positioning.

FA40 - FA54 Selecting largest value for positioning.

FA58 - FA62 Solution for ℓ_{ir} $(i = r + 1,\ldots,n)$ in the above equations.

FA64 - FA66 Printing of small pivot elements.

SYMBOL DICTIONARY

A = input transposed matrix overwritten with calculated L^T and U^T matrices

CONJG = external routine (conjugate of a complex number)

D = scratch vector

DMAX = maximum value in D

ELMAG = intermediate variable

I = DO loop index

IFLG = small pivot flag

IP = integer vector storing positioning information

J = DO loop index

JP1 = J + 1

K = DO loop index

N = order of matrix being factored

NDIM = dimensions of the array where the matrix is stored. NDIM $\geq$ N

PJ = intermediate variable

PR = intermediate variable

R	= DO loop index
REAL	= external routine (real part of complex number)
RM1	= R - 1
RP1	= R + 1

```
      SUBROUTINE FACTR (N,A,IP,NDIM)
C
C     SUBROUTINE TO FACTOR A MATRIX INTO A UNIT LOWER TRIANGULAR MATRIX
C     AND AN UPPER TRIANGULAR MATRIX USING THE GAUSS-DOOLITTLE ALGORITHM
C     PRESENTED ON PAGES 411-416 OF A. RALSTON--A FIRST COURSE IN
C     NUMERICAL ANALYSIS.  COMMENTS BELOW REFER TO COMMENTS IN RALSTONS
C     TEXT.    (MATRIX TRANSPOSED.
C
      COMPLEX A,D,ARJ
      DIMENSION A(NDIM,NDIM), IP(NDIM)
      COMMON /SCRATM/ D(600)
      INTEGER R,RM1,RP1,PJ,PR
      IFLG=0
      DO 9 R=1,N
C
C     STEP 1
C
      DO 1 K=1,N
      D(K)=A(R,K)
    1 CONTINUE
C
C     STEPS 2 AND 3
C
      RM1=R-1
      IF (RM1.LT.1) GO TO 4
      DO 3 J=1,RM1
      PJ=IP(J)
      ARJ=D(PJ)
      A(R,J)=ARJ
      D(PJ)=D(J)
      JP1=J+1
      DO 2 I=JP1,N
      D(I)=D(I)-A(J,I)*ARJ
    2 CONTINUE
    3 CONTINUE
    4 CONTINUE
C
C     STEP 4
C
      DMAX=REAL(D(R)*CONJG(D(R)))
      IP(R)=R
      RP1=R+1
      IF (RP1.GT.N) GO TO 6
      DO 5 I=RP1,N
      ELMAG=REAL(D(I)*CONJG(D(I)))
      IF (ELMAG.LT.DMAX) GO TO 5
      DMAX=ELMAG
      IP(R)=I
    5 CONTINUE
    6 CONTINUE
      IF (DMAX.LT.1.E-10) IFLG=1
      PR=IP(R)
      A(R,R)=D(PR)
      D(PR)=D(R)
C
C     STEP 5
C
      IF (RP1.GT.N) GO TO 8
      ARJ=1./A(R,R)
      DO 7 I=RP1,N
      A(R,I)=D(I)*ARJ
    7 CONTINUE
    8 CONTINUE
      IF (IFLG.EQ.0) GO TO 9
```

```
      PRINT 10, R,DMAX
      IFLG=0
    9 CONTINUE
      RETURN
C
   10 FORMAT (1H ,6HPIVOT(,I3,2H)=,E16.8)
      END
```

FACTRS

PURPOSE

To call the appropriate subroutines for the LU decomposition of a matrix.

METHOD

The operation of FACTRS depends on the mode of storage of the matrix as determined by the value of ICASE (see COMMON/MATPAR/ in Section III). For ICASE = 1 subroutine FACTR is called at FS16 to factor the matrix. For ICASE = 2 FACTR is called for each of the NOP submatrices. If ICASE = 3 FACIO and LUNSCR are called at FS23 and FS24. FACIO reads the matrix from file IU1 and writes the result on file IU2. LUNSCR leaves the final result on file IU3.

For ICASE = 4 (symmetry, submatrices fit in core) or ICASE = 5 (symmetry, submatrices do not fit in core) the matrix elements on file IU1 are written in a new order on file IU2 from FS29 to FS46. The sequence of data on file IU1 is

column 1 of submatrix 1
column 1 of submatrix 2
⋮
column 1 of submatrix NOP
column 2 of submatrix 1
⋮
column 2 of submatrix NOP
column 3 of submatrix 1
⋮
column NPBLK of submatrix NOP

The matrices are written onto file IU2 in the sequence

column 1 of submatrix 1
column 2 of submatrix 1
⋮

column NPBLK of submatrix 1

column 1 of submatrix 2

⋮

column NPBLK of submatrix NOP

For ICASE = 4 each submatrix is then read into memory at FS58 and decomposed into LU factors by calling FACTR at FS60. The factored matrices are written to file IU3 at FS61.

For ICASE = 5 the matrices are transferred from file IU2 to IU1 at FS76 to FS77. Subroutine FACIO is then called to factor all of the NOP submatrices. The result is left on file IU2. LUNSCR reorders the rows of each matrix and leaves the result on IU3.

SYMBOL DICTIONARY

A	= array for matrix storage
I2	= number of words in a block
ICOLS	= number of columns in a block
IP	= array for pivot element indices
IR1, IR2, IRR1, IRR2	= row indices for reordering columns
IU1, IU2, IU3, IU4	= file numbers
IX	= array of pivot element data
KA	= starting location of a submatrix in the array
NOP	= number of symmetric sections
NP	= number of equations for each symmetric section (order of submatrix)
NROW	= total number of equations (NP x NOP)

```
      SUBROUTINE FACTRS (NP,NROW,A,IP,IX,IU1,IU2,IU3,IU4)
C
C     FACTRS, FOR SYMMETRIC STRUCTURE, TRANSFORMS SUBMATRICIES TO FORM
C     MATRICIES OF THE SYMMETRIC MODES AND CALLS ROUTINE TO FACTOR
C     MATRICIES.  IF NO SYMMETRY, THE ROUTINE IS CALLED TO FACTOR THE
C     COMPLETE MATRIX.
C
      COMPLEX A
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      DIMENSION A(1), IP(NROW), IX(NROW)
      NOP=NROW/NP
      IF (ICASE.GT.2) GO TO 2
      DO 1 KK=1,NOP
      KA=(KK-1)*NP+1
1     CALL FACTR (NP,A(KA),IP(KA),NROW)
      RETURN
2     IF (ICASE.GT.3) GO TO 3
C
C     FACTOR SUBMATRICIES, OR FACTOR COMPLETE MATRIX IF NO SYMMETRY
C     EXISTS.
C
      CALL FACIO (A,NROW,NOP,IX,IU1,IU2,IU3,IU4)
      CALL LUNSCR (A,NROW,NOP,IP,IX,IU2,IU3,IU4)
      RETURN
C
C     REWRITE THE MATRICES BY COLUMNS ON TAPE 13
C
3     I2=2*NPBLK*NROW
      REWIND IU2
      DO 5 K=1,NOP
      REWIND IU1
      ICOLS=NPBLK
      IR2=K*NP
      IR1=IR2-NP+1
      DO 5 L=1,NBLOKS
      IF (NBLOKS.EQ.1.AND.K.GT.1) GO TO 4
      CALL BLCKIN (A,IU1,1,I2,1,602)
      IF (L.EQ.NBLOKS) ICOLS=NLAST
4     IRR1=IR1
      IRR2=IR2
      DO 5 ICOLDX=1,ICOLS
      WRITE (IU2) (A(I),I=IRR1,IRR2)
      IRR1=IRR1+NROW
      IRR2=IRR2+NROW
5     CONTINUE
      REWIND IU1
      REWIND IU2
      IF (ICASE.EQ.5) GO TO 8
      REWIND IU3
      IRR1=NP*NP
      DO 7 KK=1,NOP
      IR1=1-NP
      IR2=0
      DO 6 I=1,NP
      IR1=IR1+NP
      IR2=IR2+NP
6     READ (IU2) (A(J),J=IR1,IR2)
      KA=(KK-1)*NP+1
      CALL FACTR (NP,A,IP(KA),NP)
      WRITE (IU3) (A(I),I=1,IRR1)
7     CONTINUE
      REWIND IU2
      REWIND IU3
```

```
      RETURN                                            FS  65
    8 I2=2*NPSYM*NP                                     FS  66
      DO 10 KK=1,NOP                                    FS  67
      J2=NPSYM                                          FS  68
      DO 10 L=1,NBLSYM                                  FS  69
      IF (L.EQ.NBLSYM) J2=NLSYM                         FS  70
      IR1=1-NP                                          FS  71
      IR2=0                                             FS  72
      DO 9 J=1,J2                                       FS  73
      IR1=IR1+NP                                        FS  74
      IR2=IR2+NP                                        FS  75
    9 READ (IU2) (A(I),I=IR1,IR2)                       FS  76
   10 CALL BLCKOT (A,IU1,1,I2,1,193)                    FS  77
      REWIND IU1                                        FS  78
      CALL FACIO (A,NP,NOP,IX,IU1,IU2,IU3,IU4)          FS  79
      CALL LUNSCR (A,NP,NOP,IP,IX,IU2,IU3,IU4)          FS  80
      RETURN                                            FS  81
      END                                               FS  82-
```

FBAR

PURPOSE

To compute the Sommerfeld attenuation function for Norton's asymptotic field approximations.

METHOD

The value returned for FBAR is

$$F(P) = 1 - j\sqrt{\pi P}\exp(-P)\,[1 - \mathrm{erf}(j\sqrt{P})]$$

where erf(z) is the error function. If $|j\sqrt{P}| \leq 3$ the value of $\mathrm{erf}(j\sqrt{P})$ is computed from the series

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n z^{2n+1}}{n!\,(2n+1)}$$

For $|j\sqrt{P}| > 3$, F(P) is evaluated from the first six terms of the asymptotic expansion

$$\sqrt{\pi}\, z \exp(z^2)\,(1 - \mathrm{erf}(z)) \approx 1 + \sum_{M=1}^{\infty}(-1)^M\frac{1 \cdot 3 \ldots (2M - 1)}{(2z^2)^M}$$

for $z \to \infty$, $|\arg(z)| < \frac{3\pi}{4}$

SYMBOL DICTIONARY

ACCS = relative convergence test value

FJ = $j = \sqrt{-1}$

MINUS = 1 if $\mathrm{Re}(z) < 0$

P = P

POW = $(-1)^n z^{2n+1}/n!$

SMS = magnitude squared of series

SP = $\sqrt{\pi}$

SUM = series value

TERM	=	term in the series
TMS	=	$\lvert TERM \rvert^2$
TOSP	=	$2/\sqrt{\pi}$
Z	=	$j\sqrt{P}$
ZS	=	z^2

FBAR

```
      COMPLEX FUNCTION FBAR(P)
C
C     FBAR IS SOMMERFELD ATTENUATION FUNCTION FOR NUMERICAL DISTANCE P
C
      COMPLEX Z,ZS,SUM,POW,TERM,P,FJ
      DIMENSION FJX(2)
      EQUIVALENCE (FJ,FJX)
      DATA TOSP/1.128379167/,ACCS/1.E-12/,SP/1.772453851/,FJX/0.,1./
      Z=FJ*CSQRT(P)
      IF (CABS(Z).GT.3.) GO TO 3
C
C     SERIES EXPANSION
C
      ZS=Z*Z
      SUM=Z
      POW=Z
      DO 1 I=1,100
      POW=-POW*ZS/FLOAT(I)
      TERM=POW/(2.*I+1.)
      SUM=SUM+TERM
      TMS=REAL(TERM*CONJG(TERM))
      SMS=REAL(SUM*CONJG(SUM))
      IF (TMS/SMS.LT.ACCS) GO TO 2
1     CONTINUE
2     FBAR=1.-(1.-SUM*TOSP)*Z*CEXP(ZS)*SP
      RETURN
C
C     ASYMPTOTIC EXPANSION
C
3     IF (REAL(Z).GE.0.) GO TO 4
      MINUS=1
      Z=-Z
      GO TO 5
4     MINUS=0
5     ZS=.5/(Z*Z)
      SUM=(0.,0.)
      TERM=(1.,0.)
      DO 6 I=1,6
      TERM=-TERM*(2.*I-1.)*ZS
6     SUM=SUM+TERM
      IF (MINUS.EQ.1) SUM=SUM-2.*SP*Z*CEXP(Z*Z)
      FBAR=-SUM
      RETURN
      END
```

FBLOCK

PURPOSE

To set parameters for storage of the interaction matrix.

METHOD

FBLOCK sets values of the parameters ICASE through NLSYM in COMMON/MATPAR/. The input parameters NROW and NCOL are the number of rows and columns in the non-transposed matrix. IMAX is the number of matrix elements that can be stored in the array in COMMON/CMB/. If a NGF file will be written (WG card) then IRNGF complex locations are reserved for future use. If a NGF file has not been requested then IRNGF is zero.

If (NROW)(NCOL) $\leq$ IMAX - IRNGF the complete matrix can be stored in COMMON/CMB/. ICASE is then 1 for no symmetry or 2 for symmetry. If the structure has symmetry and one submatrix fits in core but not the complete matrix,

$$(NROW)(NCOL) > IMAX - IRNGF$$
$$NROW^2 \leq IMAX - IRNGF,$$

then ICASE is 4.

If the matrix cannot fit in core for the LU decomposition then it is divided into blocks of rows (columns of the transposed matrix) for transfer between core and file storage. The blocks are made as large as possible so that one block fits into IMAX - IRNGF locations and two blocks fit into IMAX locations. Since two blocks are needed in core only during the Gauss elimination process this makes at least IRNGF locations available during the NGF solution.

CODING

FB10 - FB17	ICASE = 1 or 2
FB20 - FB32	ICASE = 3
FB34 - FB40	ICASE = 4 or 5, block parameters for whole matrix
FB42 - FB48	ICASE = 4, block parameters for submatrices
FB49 - FB58	ICASE = 5, block parameters for submatrices

FBLOCK

FB65 - FB71	S matrix for rotational symmetry (Equation III of Part I)
FB75 - FB88	S matrix for plane symmetry

SYMBOL DICTIONARY

ARG	= $2\pi(I - 1)(J - 1)/NOP$
IMAX	= number of complex numbers that can be stored in COMMON/CMB/
IMX1	= IMAX - IRNGF
IPSYM	= parameter from COMMON/DATA/
IRNGF	= array storage reserved for NGF
KA	= number of planes of symmetry
NCOL	= number of columns in matrix
NOP	= number of symmetric sections
NROW	= number of rows in matrix
PHAZ	= $2\pi/NOP$

```
      SUBROUTINE FBLOCK (NROW,NCOL,IMAX,IRNGF,IPSYM)
C     FBLOCK SETS PARAMETERS FOR OUT-OF-CORE SOLUTION FOR THE PRIMARY
C     MATRIX (A)
      COMPLEX SSX,DETER
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      COMMON /SMAT/ SSX(16,16)
      IMX1=IMAX-IRNGF
      IF (NROW*NCOL.GT.IMX1) GO TO 2
      NBLOKS=1
      NPBLK=NROW
      NLAST=NROW
      IMAT=NROW*NCOL
      IF (NROW.NE.NCOL) GO TO 1
      ICASE=1
      RETURN
1     ICASE=2
      GO TO 5
2     IF (NROW.NE.NCOL) GO TO 3
      ICASE=3
      NPBLK=IMAX/(2*NCOL)
      NPSYM=IMX1/NCOL
      IF (NPSYM.LT.NPBLK) NPBLK=NPSYM
      IF (NPBLK.LT.1) GO TO 12
      NBLOKS=(NROW-1)/NPBLK
      NLAST=NROW-NBLOKS*NPBLK
      NBLOKS=NBLOKS+1
      NBLSYM=NBLOKS
      NPSYM=NPBLK
      NLSYM=NLAST
      IMAT=NPBLK*NCOL
      PRINT 14, NBLOKS,NPBLK,NLAST
      GO TO 11
3     NPBLK=IMAX/NCOL
      IF (NPBLK.LT.1) GO TO 12
      IF (NPBLK.GT.NROW) NPBLK=NROW
      NBLOKS=(NROW-1)/NPBLK
      NLAST=NROW-NBLOKS*NPBLK
      NBLOKS=NBLOKS+1
      PRINT 14, NBLOKS,NPBLK,NLAST
      IF (NROW*NROW.GT.IMX1) GO TO 4
      ICASE=4
      NBLSYM=1
      NPSYM=NROW
      NLSYM=NROW
      IMAT=NROW*NROW
      PRINT 15
      GO TO 5
4     ICASE=5
      NPSYM=IMAX/(2*NROW)
      NBLSYM=IMX1/NROW
      IF (NBLSYM.LT.NPSYM) NPSYM=NBLSYM
      IF (NPSYM.LT.1) GO TO 12
      NBLSYM=(NROW-1)/NPSYM
      NLSYM=NROW-NBLSYM*NPSYM
      NBLSYM=NBLSYM+1
      PRINT 16, NBLSYM,NPSYM,NLSYM
      IMAT=NPSYM*NROW
5     NOP=NCOL/NROW
      IF (NOP*NROW.NE.NCOL) GO TO 13
      IF (IPSYM.GT.0) GO TO 7
C
C     SET UP SSX MATRIX FOR ROTATIONAL SYMMETRY.
C
```

```
          PHAZ=6.2831853072/NOP
          DO 6 I=2,NOP
          DO 6 J=I,NOP
          ARG=PHAZ*FLOAT(I-1)*FLOAT(J-1)
          SSX(I,J)=CMPLX(COS(ARG),SIN(ARG))
   6      SSX(J,I)=SSX(I,J)
          GO TO 11
C
C         SET UP SSX MATRIX FOR PLANE SYMMETRY
C
   7      KK=1
          SSX(1,1)=(1.,0.)
          IF ((NOP.EQ.2).OR.(NOP.EQ.4).OR.(NOP.EQ.8)) GO TO 8
          STOP
   8      KA=NOP/2
          IF (NOP.EQ.8) KA=3
          DO 10 K=1,KA
          DO 9 I=1,KK
          DO 9 J=1,KK
          DETER=SSX(I,J)
          SSX(I,J+KK)=DETER
          SSX(I+KK,J+KK)=-DETER
   9      SSX(I+KK,J)=DETER
  10      KK=KK*2
  11      RETURN
  12      PRINT 17, NROW,NCOL
          STOP
  13      PRINT 18, NROW,NCOL
          STOP
C
  14      FORMAT (//35H MATRIX FILE STORAGE -  NO. BLOCKS=,I5,19H COLUMNS PE
         1R BLOCK=,I5,23H COLUMNS IN LAST BLOCK=,I5)
  15      FORMAT (25H SUBMATRICIES FIT IN CORE)
  16      FORMAT (38H SUBMATRIX PARTITIONING -  NO. BLOCKS=,I5,19H COLUMNS P
         1ER BLOCK=,I5,23H COLUMNS IN LAST BLOCK=,I5)
  17      FORMAT (40H ERROR - INSUFFICIENT STORAGE FOR MATRIX,2I5)
  18      FORMAT (28H SYMMETRY ERROR - NROW,NCOL=,2I5)
          END
```

FBNGF

PURPOSE

To set parameters for storage of the matrices B, C and D for the NGF solution.

METHOD

The modes of matrix storage for the NGF solution are described in Section VIII. FBNGF choses the smallest ICASX (1 through 4) possible given the size of the matrices A, B, C and D and the space available in the array CM in COMMON/CMB/. If B, C and D must be divided into blocks (ICASX = 3 or 4) the blocks are chosen are large as possible to minimize the number of input and output requests. Parameters specifying the number and size of blocks are stored in COMMON/MATPAR/ (see Section III).

FBNGF also sets the locations in CM at which storage of B, C and D start. For example, CM(IC11) is passed from the main program to subroutines CMNGF and FACGF as the starting location of array C.

SYMBOL DICTIONARY

IB11	= location in CM at which storage of B starts
IC11	= location in CM at which storage of C starts
ID11	= location in CM at which storage of D starts
IMAT	= number of complex numbers in A_F
IR	= space available (complex numbers) in CM when A_F is not being used.
IRESRV	= total length of CM
IRESX	= space availabe in CM when A_F is being used
IX11	= location in CM at which storage of B starts when $A^{-1}B$ is computed (A_F occupies space in CM)
NBCD	= number of complex numbers in B, C and D combined
NBLN	= number of complex numbers in B or C
NDLN	= length of D
NEQ	= number of rows in B, columns in C
NEQ2	= number of columns in B or D, rows in C or D

```
      SUBROUTINE FBNGF (NEQ,NEQ2,IRESRV,IB11,IC11,ID11,IX11)
C     FBNGF SETS THE BLOCKING PARAMETERS FOR THE B, C, AND D ARRAYS FOR
C     OUT-OF-CORE STORAGE.
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      IRESX=IRESRV-IMAT
      NBLN=NEQ*NEQ2
      NDLN=NEQ2*NEQ2
      NBCD=2*NBLN+NDLN
      IF (NBCD.GT.IRESX) GO TO 1
      ICASX=1
      IB11=IMAT+1
      GO TO 2
1     IF (ICASE.LT.3) GO TO 3
      IF (NBCD.GT.IRESRV.OR.NBLN.GT.IRESX) GO TO 3
      ICASX=2
      IB11=1
2     NBBX=1
      NPBX=NEQ
      NLBX=NEQ
      NBBL=1
      NPBL=NEQ2
      NLBL=NEQ2
      GO TO 5
3     IR=IRESRV
      IF (ICASE.LT.3) IR=IRESX
      ICASX=3
      IF (NDLN.GT.IR) ICASX=4
      NBCD=2*NEQ+NEQ2
      NPBL=IR/NBCD
      NLBL=IR/(2*NEQ2)
      IF (NLBL.LT.NPBL) NPBL=NLBL
      IF (ICASE.LT.3) GO TO 4
      NLBL=IRESX/NEQ
      IF (NLBL.LT.NPBL) NPBL=NLBL
4     IF (NPBL.LT.1) GO TO 6
      NBBL=(NEQ2-1)/NPBL
      NLBL=NEQ2-NBBL*NPBL
      NBBL=NBBL+1
      NBLN=NEQ*NPBL
      IR=IR-NBLN
      NPBX=IR/NEQ2
      IF (NPBX.GT.NEQ) NPBX=NEQ
      NBBX=(NEQ-1)/NPBX
      NLBX=NEQ-NBBX*NPBX
      NBBX=NBBX+1
      IB11=1
      IF (ICASE.LT.3) IB11=IMAT+1
5     IC11=IB11+NBLN
      ID11=IC11+NBLN
      IX11=IMAT+1
      PRINT 11, NEQ2
      IF (ICASX.EQ.1) RETURN
      PRINT 8, ICASX
      PRINT 9, NBBX,NPBX,NLBX
      PRINT 10, NBBL,NPBL,NLBL
      RETURN
6     PRINT 7, IRESRV,IMAT,NEQ,NEQ2
      STOP
C
7     FORMAT (55H ERROR - INSUFFICIENT STORAGE FOR INTERACTION MATRICIES
     1,24H  IRESRV,IMAT,NEQ,NEQ2 =,4I5)
8     FORMAT (48H FILE STORAGE FOR NEW MATRIX SECTIONS -   ICASX =,I2)
9     FORMAT (19H B FILLED BY ROWS -,15X,12HNO. BLOCKS =,I3,3X,16HROWS P
```

```
     1ER BLOCK =,I3,3X,20HROWS IN LAST BLOCK =,I3)                    FN  65
  10  FORMAT (32H B BY COLUMNS, C AND D BY ROWS -,2X,12HNO. BLOCKS =,I3,  FN  66
     14X,15HR/C PER BLOCK =,I3,4X,19HR/C IN LAST BLOCK =,I3)         FN  67
  11  FORMAT (//,35H N.G.F. - NUMBER OF NEW UNKNOWNS IS,I4)           FN  68
      END                                                             FN  69-
```

FFLD

PURPOSE

To calculate the radiated electric field due to the currents on wires and surfaces in free space or over ground. The range factor $\exp(-jkr_0)/(r_0/\lambda)$ is omitted.

METHOD

Equation (126) of Part I is used to evaluate the radiated field of wires and surfaces. The surface part of the equation is evaluated in subroutine FFLDS, however. For wires, the field equation is

$$\overline{E}(\overline{r}_0) = \frac{j\eta \exp(-jkr_0)}{4\pi\ r_0/\lambda} (\hat{k}\hat{k} - \overline{\overline{I}}) \cdot \overline{F}(\overline{r}_0)$$

$$\overline{F}(\overline{r}_0) = 2\pi \int_L \exp(j\overline{k}\cdot\overline{r})\ [\overline{I}(s)/\lambda]\ ds/\lambda$$

where

$r_0 = |\overline{r}_0|$

$\hat{k} = \overline{r}_0/|\overline{r}_0|$

$k = 2\pi/\lambda$

$\overline{k} = k\hat{k}$

$\overline{I}(s)$ = current on the wire at s

$\overline{\overline{I}}$ = identity dyad

L = contour of the wire

$\overline{r}$ = position of the point at s on the wire

The dot product with the dyad $\hat{k}\hat{k} - \overline{\overline{I}}$ results in the component of $\overline{F}$ transverse to $\hat{k}$. This is accomplished in the code by computing the dot products with the unit vectors $\hat{\theta}$ and $\hat{\phi}$, normal to $\hat{k}$.

For a wire structure consisting of N straight segments, $\overline{r}$ on segment i is replaced by

$$\overline{r} = \overline{r}_i + \lambda t\hat{u}_i ,$$

where

$\overline{r}_i$ = location of the center of segment i

$\overline{u}_i$ = unit vector in the direction of segment i

Then, $\overline{F}$ is evaluated as

$$\overline{F}(\overline{r}_0) = \sum_{i=1}^{N} \exp(j\overline{k}\cdot\overline{r}_i)\ \overline{Q}_i$$

$$Q_i = 2\pi\hat{u}_i \int_{-\Delta_i/2}^{\Delta_i/2} \exp[j2\pi t(\hat{k}\cdot\hat{u}_i)]\ I_i(t)/\lambda\ dt$$

where Δ_i is the length of segment i normalized to λ. With

$$I_i(t)/\lambda = A_i + B_i \sin(2\pi t) + C_i \cos(2\pi t)\ ,$$

the integral can be evaluated as

$$\overline{Q}_i = \hat{u}_i \left\{ A_i \frac{2\sin(\pi w_i\Delta_i)}{w_i} - jB_i\left[\frac{\sin[\pi(1-w_i)\Delta_i]}{(1-w_i)} - \frac{\sin[\pi(1+w_i)\Delta_i]}{(1+w_i)}\right] + C_i\left[\frac{\sin[\pi(1-w_i)\Delta_i]}{(1-w_i)} + \frac{\sin[\pi(1+w_i)\Delta_i]}{(1+w_i)}\right]\right\},$$

where $w_i = -\hat{k}\cdot\hat{u}_i$.

The effect of a ground is included by computing the field of the image of each segment and modifying it by the Fresnel reflection coefficients. The coding here differs from section II-4 of Part I in some respects. Rather than reflecting each segment in the ground plane, the direction of observation, $\hat{k}$, is reflected for the image calculation. Thus, the sign of the z component of $\hat{k}$ is changed at the start of the image calculation. The z component of the image field must also be changed in sign at the end of the calculation. Also, the change in sign of the image field due to the change in sign of charge on the image is combined with the reflection coefficients. Thus, the reflection coefficients are the negative of those in Part I.

The code allows for a change in ground height and electrical parameters at a fixed radial distance from the origin (circular cliff) or at a fixed distance in x (linear cliff). In these cases, the reflection point of the ray from the center of each segment is computed, and the reflection coefficients and phase lag are computed for the appropriate ground. Effects from the region of change, such as diffraction from the edge, are not included,

however. A radial wire ground screen may also be included by the reflection coefficient approximation described in section II-4 of Part I.

CODING

FF30 - FF164	Calculation of field due to segments.
FF34 - FF164	Loop over direct and image fields.
FF38 - FF63	Reflection coefficients computed.
FF64	$\hat{k}$ reflected in ground for image.
FF65 - FF70	Direct fields saved, and CIX, CIY, CIZ initialized before image calculation.
FF75 - FF96	Field of segment I computed.
FF102 - FF104	Summation of fields for direct field or uniform ground.
FF110 - FF149	Appropriate reflection coefficient determined and field summed for reflected field from two-medium ground or radial-wire ground screen.
FF156 - FF159	Image field multiplied by reflection coefficients for uniform ground and added to direct field.
FF161 - FF163	Reflected field added to direct field for two-medium ground or radial wire ground.
FF166 - FF167	Dot products of $\overline{F}$ with $\hat{\theta}$ and $\hat{\phi}$ for wires only.
FF169 - FF208	Calculation of field due to surface patches.
FF177 - FF203	Loop over direct and image fields.
FF179	$\hat{k}$ reflected for image.
FF180	FFLDS calculates field.
FF186 - FF202	Field multiplied by reflection coefficients for uniform ground only.

SYMBOL DICTIONARY

A	$= 2 \sin (\pi w_i \Delta_i)/w_i$ (a series is used for small w_i)
ARG	$= \overline{k} \cdot \overline{r}_i$
B	= coefficient of B_i in $\overline{Q}_i$
BOO	$= \sin [\pi(1 - w_i)\Delta_i]/[\pi(1 - w_i)\Delta_i]$
BOT	$= \pi(1 - w_i)\Delta_i$
C	= coefficient of C_i in $\overline{Q}_i$
CAB, SAB, SALP	= x, y, z components of $\hat{u}_i$

CCX, CCY, CCZ = variables for summation of x, y, and z components of $\overline{F}$

CDP = $(\overline{F} \cdot \hat{\phi})(R_V - R_H)$

CIX, CIY, CIZ = variables for summation of x, y, and z components of $\overline{F}$

CONST = CONSX = $-j\eta/4\pi$

D = distance of ray reflection point from origin

DARG = phase increment due to change in ground level

EL = $\pi\Delta_i$

EPH = ϕ component of $(r_0/\lambda)\exp(jkr_0)\ \overline{E}(\overline{r}_0)$

ETH = θ component of $(r_0/\lambda)\exp(jkr_0)\ \overline{E}(\overline{r}_0)$

ETA = $\eta = \sqrt{\mu/\varepsilon}$

EX, EY, EZ = $(r_0/\lambda)\exp(jkr_0)\ \overline{E}(\overline{r}_0)$ for patches

EXA = Q_i

GX, GY, GZ = $(r_0/\lambda)\exp(jkr_0)\ \overline{E}(\overline{r}_0)$ for direct and reflected fields of patches

I = segment number

OMEGA = w_i

PHI = ϕ

PHX, PHY = x and y components of $\hat{\phi}$

PI = π

RFL = ±1 for direct or image field of patch

RI = imaginary part of Q_i

ROX, ROY, ROZ = x, y, and z components of $\hat{k}$

ROZS = saved value of ROZ

RR = real part of Q_i

RRH = $-R_H$

RRH1 = $-R_H$ for first ground medium

RRH2 = $-R_H$ for second ground medium

RRV	$= -R_V$
RRV1	$= -R_V$ for first ground medium
RRV2	$= -R_V$ for second ground medium
RRZ	= z component of $\hat{k}$
SILL	$= \pi w_i \Delta_i$
THET	$= \theta$ (angle from vertical to $\hat{k}$)
THX, THY, THZ	$= \hat{\theta}$
TIX, TIY, TIZ	$= Q_i$ for image in ground
TOO	$= \sin[\pi(1 + w_i)\Delta_i]/[\pi(1 + w_i)\Delta_i]$
TOP	$= \pi(1 + w_i)\Delta_i$
TP	$= 2\pi$
TTHET	$= \tan \theta$
ZRATI	$= [\varepsilon_r - j\sigma/(\omega\varepsilon_0)]^{-1/2}$ ε_r, σ = ground parameters
ZRSIN	$= [1 - (\text{ZRATI})^2 \sin^2 \theta]^{1/2}$
ZSCRN	= surface impedance of ground with radial wire ground screen

CONSTANTS

-29.97922085	$= -j\eta/(4\pi)$
3.141592654	$= \pi$
376.73	$= \eta$
6.283185308	$= 2\pi$

```
      SUBROUTINE FFLD (THET,PHI,ETH,EPH)
C
C     FFLD CALCULATES THE FAR ZONE RADIATED ELECTRIC FIELDS.
C     THE FACTOR EXP(J*K*R)/(R/LAMDA) NOT INCLUDED
C
      COMPLEX CIX,CIY,CIZ,EXA,ETH,EPH,CONST,CCX,CCY,CCZ,CDP,CUR
      COMPLEX ZRATI,ZRSIN,RRV,RRH,RRV1,RRH1,RRV2,RRH2,ZRATI2,TIX,TIY,TIZ
     1,T1,ZSCRN,EX,EY,EZ,GX,GY,GZ,FRATI
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),CII(300
     1),CUR(900)
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      DIMENSION CAB(1), SAB(1), CONSX(2)
      EQUIVALENCE (CAB,ALP), (SAB,BET), (CONST,CONSX)
      DATA PI,TP,ETA/3.141592654,6.283185308,376.73/
      DATA CONSX/0.,-29.97922085/
      PHX=-SIN(PHI)
      PHY=COS(PHI)
      ROZ=COS(THET)
      ROZS=ROZ
      THX=ROZ*PHY
      THY=-ROZ*PHX
      THZ=-SIN(THET)
      ROX=-THZ*PHY
      ROY=THZ*PHX
      IF (N.EQ.0) GO TO 20
C
C     LOOP FOR STRUCTURE IMAGE IF ANY
C
      DO 19 K=1,KSYMP
C
C     CALCULATION OF REFLECTION COEFFECIENTS
C
      IF (K.EQ.1) GO TO 4
      IF (IPERF.NE.1) GO TO 1
C
C     FOR PERFECT GROUND
C
      RRV=-(1.,0.)
      RRH=-(1.,0.)
      GO TO 2
C
C     FOR INFINITE PLANAR GROUND
C
1     ZRSIN=CSQRT(1.-ZRATI*ZRATI*THZ*THZ)
      RRV=-(ROZ-ZRATI*ZRSIN)/(ROZ+ZRATI*ZRSIN)
      RRH=(ZRATI*ROZ-ZRSIN)/(ZRATI*ROZ+ZRSIN)
2     IF (IFAR.LE.1) GO TO 3
C
C     FOR THE CLIFF PROBLEM, TWO REFLCTION COEFFICIENTS CALCULATED
C
      RRV1=RRV
      RRH1=RRH
      TTHET=TAN(THET)
      IF (IFAR.EQ.4) GO TO 3
      ZRSIN=CSQRT(1.-ZRATI2*ZRATI2*THZ*THZ)
      RRV2=-(ROZ-ZRATI2*ZRSIN)/(ROZ+ZRATI2*ZRSIN)
      RRH2=(ZRATI2*ROZ-ZRSIN)/(ZRATI2*ROZ+ZRSIN)
      DARG=-TP*2.*CH*ROZ
3     ROZ=-ROZ
```

```
      CCX=CIX
      CCY=CIY
      CCZ=CIZ
4     CIX=(0.,0.)
      CIY=(0.,0.)
      CIZ=(0.,0.)
C
C     LOOP OVER STRUCTURE SEGMENTS
C
      DO 17 I=1,N
      OMEGA=-(ROX*CAB(I)+ROY*SAB(I)+ROZ*SALP(I))
      EL=PI*SI(I)
      SILL=OMEGA*EL
      TOP=EL+SILL
      BOT=EL-SILL
      IF (ABS(OMEGA).LT.1.E-7) GO TO 5
      A=2.*SIN(SILL)/OMEGA
      GO TO 6
5     A=(2.-OMEGA*OMEGA*EL*EL/3.)*EL
6     IF (ABS(TOP).LT.1.E-7) GO TO 7
      TOO=SIN(TOP)/TOP
      GO TO 8
7     TOO=1.-TOP*TOP/6.
8     IF (ABS(BOT).LT.1.E-7) GO TO 9
      BOO=SIN(BOT)/BOT
      GO TO 10
9     BOO=1.-BOT*BOT/6.
10    B=EL*(BOO-TOO)
      C=EL*(BOO+TOO)
      RR=A*AIR(I)+B*BII(I)+C*CIR(I)
      RI=A*AII(I)-B*BIR(I)+C*CII(I)
      ARG=TP*(X(I)*ROX+Y(I)*ROY+Z(I)*ROZ)
      IF (K.EQ.2.AND.IFAR.GE.2) GO TO 11
      EXA=CMPLX(COS(ARG),SIN(ARG))*CMPLX(RR,RI)
C
C     SUMMATION FOR FAR FIELD INTEGRAL
C
      CIX=CIX+EXA*CAB(I)
      CIY=CIY+EXA*SAB(I)
      CIZ=CIZ+EXA*SALP(I)
      GO TO 17
C
C     CALCULATION OF IMAGE CONTRIBUTION IN CLIFF AND GROUND SCREEN
C     PROBLEMS.
C
11    DR=Z(I)*TTHET
C
C     SPECULAR POINT DISTANCE
C
      D=DR*PHY+X(I)
      IF (IFAR.EQ.2) GO TO 13
      D=SQRT(D*D+(Y(I)-DR*PHX)**2)
      IF (IFAR.EQ.3) GO TO 13
      IF ((SCRWL-D).LT.0.) GO TO 12
C
C     RADIAL WIRE GROUND SCREEN REFLECTION COEFFICIENT
C
      D=D+T2
      ZSCRN=T1*D*ALOG(D/T2)
      ZSCRN=(ZSCRN*ZRATI)/(ETA*ZRATI+ZSCRN)
      ZRSIN=CSQRT(1.-ZSCRN*ZSCRN*THZ*THZ)
      RRV=(ROZ+ZSCRN*ZRSIN)/(-ROZ+ZSCRN*ZRSIN)
      RRH=(ZSCRN*ROZ+ZRSIN)/(ZSCRN*ROZ-ZRSIN)
      GO TO 16
```

```
12    IF (IFAR.EQ.4) GO TO 14
      IF (IFAR.EQ.5) D=DR*PHY+X(I)
13    IF ((CL-D).LE.0.) GO TO 15
14    RRV=RRV1
      RRH=RRH1
      GO TO 16
15    RRV=RRV2
      RRH=RRH2
      ARG=ARG+DARG
16    EXA=CMPLX(COS(ARG),SIN(ARG))*CMPLX(RR,RI)
C
C     CONTRIBUTION OF EACH IMAGE SEGMENT MODIFIED BY REFLECTION COEF.
C     FOR CLIFF AND GROUND SCREEN PROBLEMS
C
      TIX=EXA*CAB(I)
      TIY=EXA*SAB(I)
      TIZ=EXA*SALP(I)
      CDP=(TIX*PHX+TIY*PHY)*(RRH-RRV)
      CIX=CIX+TIX*RRV+CDP*PHX
      CIY=CIY+TIY*RRV+CDP*PHY
      CIZ=CIZ-TIZ*RRV
17    CONTINUE
      IF (K.EQ.1) GO TO 19
      IF (IFAR.GE.2) GO TO 18
C
C     CALCULATION OF CONTRIBUTION OF STRUCTURE IMAGE FOR INFINITE GROUND
C
      CDP=(CIX*PHX+CIY*PHY)*(RRH-RRV)
      CIX=CCX+CIX*RRV+CDP*PHX
      CIY=CCY+CIY*RRV+CDP*PHY
      CIZ=CCZ-CIZ*RRV
      GO TO 19
18    CIX=CIX+CCX
      CIY=CIY+CCY
      CIZ=CIZ+CCZ
19    CONTINUE
      IF (M.GT.0) GO TO 21
      ETH=(CIX*THX+CIY*THY+CIZ*THZ)*CONST
      EPH=(CIX*PHX+CIY*PHY)*CONST
      RETURN
20    CIX=(0.,0.)
      CIY=(0.,0.)
      CIZ=(0.,0.)
21    ROZ=ROZS
C
C     ELECTRIC FIELD COMPONENTS
C
      RFL=-1.
      DO 25 IP=1,KSYMP
      RFL=-RFL
      RRZ=ROZ*RFL
      CALL FFLDS (ROX,ROY,RRZ,CUR(N+1),GX,GY,GZ)
      IF (IP.EQ.2) GO TO 22
      EX=GX
      EY=GY
      EZ=GZ
      GO TO 25
22    IF (IPERF.NE.1) GO TO 23
      GX=-GX
      GY=-GY
      GZ=-GZ
      GO TO 24
23    RRV=CSQRT(1.-ZRATI*ZRATI*THZ*THZ)
      RRH=ZRATI*ROZ
```

```
      RRH=(RRH-RRV)/(RRH+RRV)
      RRV=ZRATI*RRV
      RRV=-(ROZ-RRV)/(ROZ+RRV)
      ETH=(GX*PHX+GY*PHY)*(RRH-RRV)
      GX=GX*RRV+ETH*PHX
      GY=GY*RRV+ETH*PHY
      GZ=GZ*RRV
   24 EX=EX+GX
      EY=EY+GY
      EZ=EZ-GZ
   25 CONTINUE
      EX=EX+CIX*CONST
      EY=EY+CIY*CONST
      EZ=EZ+CIZ*CONST
      ETH=EX*THX+EY*THY+EZ*THZ
      EPH=EX*PHX+EY*PHY
      RETURN
      END
```

FFLDS

PURPOSE

To calculate the x, y, z components of the far electric field due to surface currents. The term $\exp(-jkr_0)/(r_0/\lambda)$ is omitted.

METHOD

The field is computed using the surface portion of equation (126) in Part I. With lengths normalized to the wavelength, the equation is

$$\overline{E}(\overline{r}_0) = \frac{j\eta}{2} \frac{\exp(-jkr_0)}{r_0/\lambda} (\hat{k}\hat{k} - \overline{\overline{I}}) \cdot \int_S \overline{J}_S(\overline{r}) \exp(j\overline{k} \cdot \overline{r}) \, dA/\lambda^2 ,$$

where

$r_0 = |\overline{r}_0|$
$\hat{k} = \overline{r}_0/|\overline{r}_0|$
$k = 2\pi/\lambda$
$\overline{k} = k\hat{k}$
$\overline{J}_S$ = surface current on surface S
$\overline{\overline{I}}$ = identity dyad

The dot product with the dyad $\hat{k}\hat{k} - \overline{\overline{I}}$ results in the component of the integral

$$\overline{F}(\overline{r}_0) = \int_S \overline{J}_S(\overline{r}) \exp(j\overline{k} \cdot \overline{r}) \, dA/\lambda^2$$

transverse to $\hat{k}$. The integral is evaluated by summation over the patches with the current assumed constant over each patch.

SYMBOL DICTIONARY

ARG = $\overline{k} \cdot \overline{r}_i$, $\overline{r}_i$ = center of patch I
CONS = CONSX = $j\eta/2$
CT = $\exp(j\overline{k} \cdot \overline{r}_i) \, dA/\lambda^2$ at FL18
= $\hat{k} \cdot \overline{F}(\overline{r}_0)$ at FL24
EX, EY, EZ = x, y, z components of $\overline{F}(\overline{r}_0)$ at FL22
= $(r_0/\lambda)\exp(jkr_0)\overline{E}(\overline{r}_0)$ at FL27
I = array location of patch data
J = patch number
K = current array index

FFLDS

ROX, ROY, ROZ = x, y, z components of $\hat{k}$

S(I) = (area of patch I)/λ^2

SCUR = array containing surface current components

TPI = 2π

XS, YS, ZS = arrays containing center point coordinates of patches normalized to wavelength.

CODE LISTING

```
      SUBROUTINE FFLDS (ROX,ROY,ROZ,SCUR,EX,EY,EZ)                      FL   1
C     CALCULATES THE XYZ COMPONENTS OF THE ELECTRIC FIELD DUE TO        FL   2
C     SURFACE CURRENTS                                                  FL   3
      COMPLEX CT,CONS,SCUR,EX,EY,EZ                                     FL   4
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 FL   5
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( FL   6
     2300),WLAM,IPSYM                                                   FL   7
      DIMENSION XS(1), YS(1), ZS(1), S(1), SCUR(1), CONSX(2)            FL   8
      EQUIVALENCE (XS,X), (YS,Y), (ZS,Z), (S,BI), (CONS,CONSX)          FL   9
      DATA TPI/6.283185308/,CONSX/0.,188.365/                           FL  10
      EX=(0.,0.)                                                        FL  11
      EY=(0.,0.)                                                        FL  12
      EZ=(0.,0.)                                                        FL  13
      I=LD+1                                                            FL  14
      DO 1 J=1,M                                                        FL  15
      I=I-1                                                             FL  16
      ARG=TPI*(ROX*XS(I)+ROY*YS(I)+ROZ*ZS(I))                           FL  17
      CT=CMPLX(COS(ARG)*S(I),SIN(ARG)*S(I))                             FL  18
      K=3*J                                                             FL  19
      EX=EX+SCUR(K-2)*CT                                                FL  20
      EY=EY+SCUR(K-1)*CT                                                FL  21
      EZ=EZ+SCUR(K)*CT                                                  FL  22
    1 CONTINUE                                                          FL  23
      CT=ROX*EX+ROY*EY+ROZ*EZ                                           FL  24
      EX=CONS*(CT*ROX-EX)                                               FL  25
      EY=CONS*(CT*ROY-EY)                                               FL  26
      EZ=CONS*(CT*ROZ-EZ)                                               FL  27
      RETURN                                                            FL  28
      END                                                               FL  29-
```

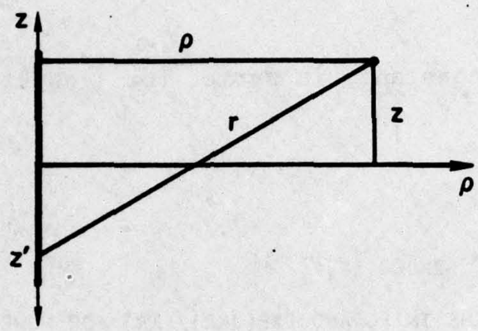
GF

PURPOSE

To supply values of the integrated function exp(jkr)/(kr) to the numerical integration routine INTX.

METHOD

The geometry parameters for integration over a segment are shown in the following diagram.



in which

$$r(z') = [\rho^2 + (z' - z)^2]^{1/2} .$$

If the field point (ρ,z) is not on the source segment, the integrand value is

$$G(z') = \frac{\exp[jkr(z')]}{kr(z')} .$$

If the field point is on the source segment (ρ = 0, z = 0), the integrand value is

$$G(z') = \frac{\exp[jkr(z')] - 1}{kr(z')} .$$

In the latter case, if kr is less than 0.2, then (cos kr)/kr is evaluated by the first three terms of its Taylor's series to reduce numerical error.

SYMBOL DICTIONARY

CO = real part of G(z')
COS = external function (cosine)
IJ = flag to indicate when field point is on source segment (by IJ = 0)
RK = kr

RKB2 = $(k\rho)^2$

SI = imaginary part of G(z')

SIN = external function (sine)

SQRT = external function (square root)

ZDK = kz' - kz

ZK = kz'

ZPK = kz

CONSTANTS

$$\left.\begin{array}{l} -1.38888889\text{E}-3 \\ 4.16666667\text{E}-2 \\ 0.5 \end{array}\right\} = \text{constants in series for } (\cos kr - 1)/kr$$

CODE LISTING

```
      SUBROUTINE GF (ZK,CO,SI)
C
C     GF COMPUTES THE INTEGRAND EXP(JKR)/(KR) FOR NUMERICAL INTEGRATION.
C
      COMMON /TMI/ ZPK,RKB2,IJ
      ZDK=ZK-ZPK
      RK=SQRT(RKB2+ZDK*ZDK)
      SI=SIN(RK)/RK
      IF (IJ) 1,2,1
    1 CO=COS(RK)/RK
      RETURN
    2 IF (RK.LT..2) GO TO 3
      CO=(COS(RK)-1.)/RK
      RETURN
    3 RKS=RK*RK
      CO=((-1.38888889E-3*RKS+4.16666667E-2)*RKS-.5)*RK
      RETURN
      END
```

PURPOSE

To read the NGF file and store parameters in the proper arrays.

METHOD

GI22	Miscellaneous parameters are read.
GI30 - GI48	Segment coordinates were converted to the form involving the segment center, segment length, and orientation (see Section III, COMMON/DATA/) with dimensions of wavelength. They must be converted back to the coordinates of the segment ends so that subroutine CONNECT can locate connections. Dimensions are converted to meters.
GI52 - GI62	Patch coordinates are converted from units of wavelength to meters since they will be scaled back to wavelengths along with the new segments and patches.
GI63	Matrix blocking parameters are read.
GI64	Interpolation tables for the Sommerfeld integrals are read if the Sommerfeld/Norton ground treatment was used.
GI74	Matrix A_F is read for in-core storage (ICASE = 1 or 2).
GI78 - GI81	A_F is read for ICASE = 4.
GI83 - GI88	A_F is read for ICASE = 3 or 5.
GI92 - GI113	A heading summarizing the NGF file is printed.

SYMBOL DICTIONARY

DX	= half segment length (meters)
IGFL	= file number for NGF file
IOUT	= number of elements in matrix
IPRT	= 1 to print coordinates of ends of segments
NBL2	= two times number of blocks in matrix A_F (since A_F is stored twice, in ascending and descending order)
NEQ	= order of the NGF matrix
NOP	= number of symmetric sections
NPEQ	= number of unknowns for a symmetric section
XI, YI, ZI	= coordinates of the center of a segment or patch

GFIL

```
      SUBROUTINE GFIL (IPRT)
C
C     GFIL READS THE N.G.F. FILE
C
      COMPLEX CM,SSX,ZRATI,ZRATI2,T1,ZARRAY,AR1,AR2,AR3,EPSCF,FRATI
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /CMB/ CM(4000)
      COMMON /ANGL/ SALP(300)
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      COMMON /GGRID/ AR1(11,10,4),AR2(17,5,4),AR3(9,8,4),EPSCF,DXA(3),DY
     1A(3),XSA(3),YSA(3),NXA(3),NYA(3)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      COMMON /SMAT/ SSX(16,16)
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF
      COMMON /SAVE/ IP(600),KCOM,COM(13,5),EPSR,SIG,SCRWLT,SCRWRT,FMHZ
      DATA IGFL/20/
      REWIND IGFL
      READ (IGFL) N1,NP,M1,MP,WLAM,FMHZ,IPSYM,KSYMP,IPERF,NRADL,EPSR,SIG
     1,SCRWLT,SCRWRT,NLODF,KCOM
      N=N1
      M=M1
      N2=N1+1
      M2=M1+1
      IF (N1.EQ.0) GO TO 2
C     READ SEG. DATA AND CONVERT BACK TO END COORD. IN UNITS OF METERS
      READ (IGFL) (X(I),I=1,N1),(Y(I),I=1,N1),(Z(I),I=1,N1)
      READ (IGFL) (SI(I),I=1,N1),(BI(I),I=1,N1),(ALP(I),I=1,N1)
      READ (IGFL) (BET(I),I=1,N1),(SALP(I),I=1,N1)
      READ (IGFL) (ICON1(I),I=1,N1),(ICON2(I),I=1,N1)
      READ (IGFL) (ITAG(I),I=1,N1)
      IF (NLODF.NE.0) READ (IGFL) (ZARRAY(I),I=1,N1)
      DO 1 I=1,N1
      XI=X(I)*WLAM
      YI=Y(I)*WLAM
      ZI=Z(I)*WLAM
      DX=SI(I)*.5*WLAM
      X(I)=XI-ALP(I)*DX
      Y(I)=YI-BET(I)*DX
      Z(I)=ZI-SALP(I)*DX
      SI(I)=XI+ALP(I)*DX
      ALP(I)=YI+BET(I)*DX
      BET(I)=ZI+SALP(I)*DX
      BI(I)=BI(I)*WLAM
1     CONTINUE
2     IF (M1.EQ.0) GO TO 4
      J=LD-M1+1
C     READ PATCH DATA AND CONVERT TO METERS
      READ (IGFL) (X(I),I=J,LD),(Y(I),I=J,LD),(Z(I),I=J,LD)
      READ (IGFL) (SI(I),I=J,LD),(BI(I),I=J,LD),(ALP(I),I=J,LD)
      READ (IGFL) (BET(I),I=J,LD),(SALP(I),I=J,LD)
      READ (IGFL) (ICON1(I),I=J,LD),(ICON2(I),I=J,LD)
      READ (IGFL) (ITAG(I),I=J,LD)
      DX=WLAM*WLAM
      DO 3 I=J,LD
      X(I)=X(I)*WLAM
      Y(I)=Y(I)*WLAM
      Z(I)=Z(I)*WLAM
3     BI(I)=BI(I)*DX
4     READ (IGFL) ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT
      IF (IPERF.EQ.2) READ (IGFL) AR1,AR2,AR3,EPSCF,DXA,DYA,XSA,YSA,NXA,
```

```
     1NYA
      NEQ=N1+2*M1
      NPEQ=NP+2*MP
      NOP=NEQ/NPEQ
      IF (NOP.GT.1) READ (IGFL) ((SSX(I,J),I=1,NOP),J=1,NOP)
      READ (IGFL) (IP(I),I=1,NEQ),COM
C     READ MATRIX A AND WRITE TAPE13 FOR OUT OF CORE
      IF (ICASE.GT.2) GO TO 5
      IOUT=NEQ*NPEQ
      READ (IGFL) (CM(I),I=1,IOUT)
      GO TO 10
5     REWIND 13
      IF (ICASE.NE.4) GO TO 7
      IOUT=NPEQ*NPEQ
      DO 6 K=1,NOP
      READ (IGFL) (CM(J),J=1,IOUT)
6     WRITE (13) (CM(J),J=1,IOUT)
      GO TO 9
7     IOUT=NPSYM*NPEQ*2
      NBL2=2*NBLSYM
      DO 8 IOP=1,NOP
      DO 8 I=1,NBL2
      CALL BLCKIN (CM,IGFL,1,IOUT,1,206)
8     CALL BLCKOT (CM,13,1,IOUT,1,205)
9     REWIND 13
10    REWIND IGFL
C     PRINT N.G.F. HEADING
      PRINT 16
      PRINT 14
      PRINT 14
      PRINT 17
      PRINT 18, N1,M1
      IF (NOP.GT.1) PRINT 19, NOP
      PRINT 20, IMAT,ICASE
      IF (ICASE.LT.3) GO TO 11
      NBL2=NEQ*NPEQ
      PRINT 21, NBL2
11    PRINT 22, FMHZ
      IF (KSYMP.EQ.2.AND.IPERF.EQ.1) PRINT 23
      IF (KSYMP.EQ.2.AND.IPERF.EQ.0) PRINT 27
      IF (KSYMP.EQ.2.AND.IPERF.EQ.2) PRINT 28
      IF (KSYMP.EQ.2.AND.IPERF.NE.1) PRINT 24, EPSR,SIG
      PRINT 17
      DO 12 J=1,KCOM
12    PRINT 15, (COM(I,J),I=1,13)
      PRINT 17
      PRINT 14
      PRINT 14
      PRINT 16
      IF (IPRT.EQ.0) RETURN
      PRINT 25
      DO 13 I=1,N1
13    PRINT 26, I,X(I),Y(I),Z(I),SI(I),ALP(I),BET(I)
      RETURN
C
14    FORMAT (5X,50H**************************************************,3
     14H**********************************)
15    FORMAT (5X,3H** ,13A6,3H **)
16    FORMAT (////)
17    FORMAT (5X,2H**,80X,2H**)
18    FORMAT (5X,29H** NUMERICAL GREEN'S FUNCTION,53X,2H**,/,5X,17H** NO
     1. SEGMENTS =,I4,10X,13HNO. PATCHES =,I4,34X,2H**)
19    FORMAT (5X,27H** NO. SYMMETRIC SECTIONS =,I4,51X,2H**)
20    FORMAT (5X,34H** N.G.F. MATRIX -  CORE STORAGE =,I7,23H COMPLEX NU
```

```
     1MBERS,  CASE,I2,16X,2H**)                                          GI 129
 21   FORMAT (5X,2H**,19X,13HMATRIX SIZE =,I7,16H COMPLEX NUMBERS,25X,2H GI 130
     1**)                                                                GI 131
 22   FORMAT (5X,14H** FREQUENCY =,E12.5,5H MHZ.,51X,2H**)              GI 132
 23   FORMAT (5X,17H** PERFECT GROUND,65X,2H**)                         GI 133
 24   FORMAT (5X,44H** GROUND PARAMETERS - DIELECTRIC CONSTANT =,E12.5,2 GI 134
     16X,2H**,/,5X,2H**,21X,14HCONDUCTIVITY =,E12.5,8H MHOS/M.,25X,2H**) GI 135
 25   FORMAT (39X,31HNUMERICAL GREEN'S FUNCTION DATA,/,41X,27HCOORDINATE GI 136
     1S OF SEGMENT ENDS,/,51X,8H(METERS),/,5X,4HSEG.,11X,19H- - - END ON GI 137
     2E - - -,26X,19H- - - END TWO - - -,/,6X,3HNO.,6X,1HX,14X,1HY,14X,1 GI 138
     3HZ,14X,1HX,14X,1HY,14X,1HZ)                                       GI 139
 26   FORMAT (1X,I7,6E15.6)                                              GI 140
 27   FORMAT (5X,55H** FINITE GROUND.  REFLECTION COEFFICIENT APPROXIMAT GI 141
     1ION,27X,2H**)                                                      GI 142
 28   FORMAT (5X,38H** FINITE GROUND.  SOMMERFELD SOLUTION,44X,2H**)    GI 143
      END                                                                GI 144-
```

GFLD

PURPOSE

To compute the electric field at intermediate distances from a radiating structure over ground, including the surface-wave field component.

METHOD

Approximate expressions for the field of a horizontal or vertical current element over a ground plane were derived by K. A. Norton (ref. 2). These expressions are used to evaluate the field of each segment in a structure and the components summed for the total field of the structure. To evaluate Norton's expressions for segment i, a local coordinate system (x', y', z') is defined (fig. 6a) with origin on the ground plane and the vertical z axis passing through segment i. In the (x, y, z) coordinate system (fig. 6 b) the location and orientation of segment i are

$$\bar{r}_i = x_i\hat{x} + y_i\hat{y} + z_i\hat{z}$$

$$\hat{i} = \cos\alpha\cos\beta\,\hat{x} + \cos\alpha\sin\beta\,\hat{y} + \sin\alpha\,\hat{z}$$

and the field observation point is at (ρ, ϕ, z). The origin of the primed coordinate system is at $(x_i, y_i, 0)$ in the umprimed coordinates, and the x' axis is along the projection of the segment on the ground plane.

Norton's expressions give the electric field in ρ', ϕ', and z' components for infinitesimal current elements either vertical or horizontal, and directed along the x' axis. To evaluate the field of a segment, the segment current is decomposed into horizontal and vertical components, and the fields of the infinitesimal current elements are integrated over the segment. Each field component for the infinitesimal current element has the form

$$E_\Delta(\rho', \phi', z') = F_1(\rho', \phi', z')\exp(-jkR_1) + F_2(\rho', \phi', z')\exp(-jkR_2) ,$$

for

$$R_1 = |\bar{R}_1|$$

$$R_2 = |\bar{R}_2|$$

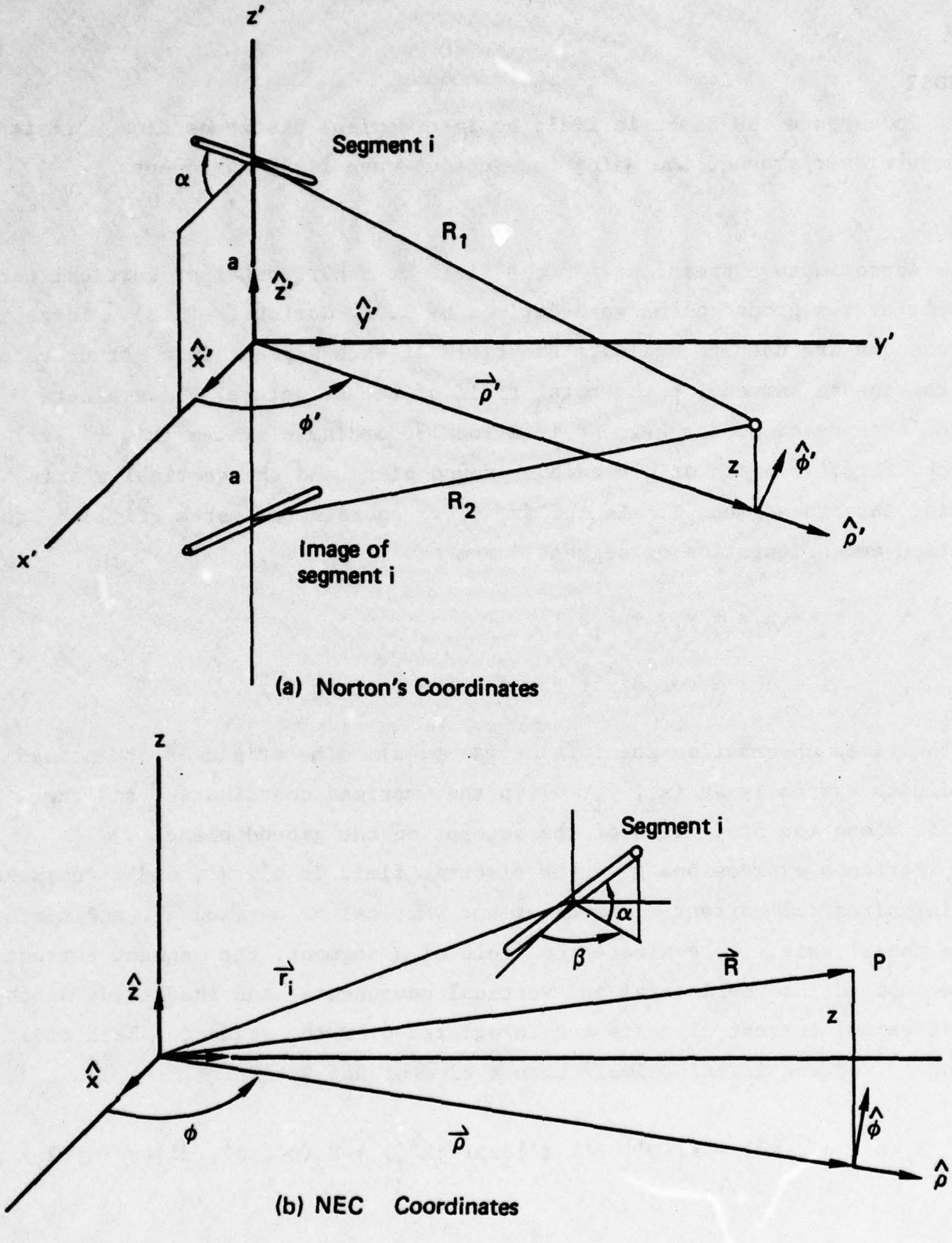


Figure 6. Coordinate Systems Used to Evaluate Norton's Expressions for the Ground Wave Fields in the NEC Program.

where F_1 and F_2 are algebraic functions of R_1 and R_2 and can be considered constant for integration over the segment as long as R_1 and R_2 are much greater than the segment length. To integrate the exponential factors over the segment, R_1 and R_2 are approximated as

$$R_1 \approx R - \hat{R}_1 \cdot (\bar{r}_i + \hat{i}s)$$

$$R_2 \approx R - \hat{R}_2 \cdot (\bar{r}'_i + \hat{i}'s)$$

where $R = |\bar{R}|$, $\hat{R}_1 = \bar{R}_1/|\bar{R}_1|$, $\hat{R}_2 = \bar{R}_2/|\bar{R}_2|$; $\bar{r}'_i$, $\hat{i}'$ = position and orientation of image of segment i, and s = variable of length along the segment (s = 0 at segment center). The current on the segment is

$$I_i(s) = A_i + B_i \sin ks + C_i \cos ks\,.$$

With F_1 and F_2 considered constant, each vector component of the field produced by segment i involves an integral of the form

$$E = F'_1 \int_{-\Delta/2\lambda}^{\Delta/2\lambda} \frac{I_i(s)}{\lambda} \exp(-jks\omega)\, d\,\frac{s}{\lambda} + F'_2 \int_{-\Delta/2\lambda}^{\Delta/2\lambda} \frac{I_i(s)}{\lambda} \exp(-jks\omega')\, d(s/\lambda)\,,$$

where

$$F'_1 = \lambda^2 F_1 \exp[-jk(R - \hat{R}_1 \cdot \bar{r}_i)]$$

$$F'_2 = \lambda^2 F_2 \exp[-jk(R - \hat{R}_2 \cdot \bar{r}'_i)]$$

$$\omega = -\hat{R}_1 \cdot \hat{i}$$

$$\omega' = -\hat{R}_2 \cdot \hat{i}'$$

Δ = segment length

The integrals can be evaluated as

$$G_1 = \int_{-\Delta/2\lambda}^{\Delta/2\lambda} \frac{I_i(s)}{\lambda} \exp(-j2\pi\, \omega\, s/\lambda)\, d\,\frac{s}{\lambda}$$

$$2\pi G_1 = \frac{A_i}{\lambda} \frac{2 \sin \pi \omega d}{\omega}$$

$$- j \frac{B_i}{\lambda} \left\{ \frac{\sin [\pi (1 - \omega)d]}{(1 - \omega)} - \frac{\sin [\pi (1 + \omega)d]}{(1 + \omega)} \right\}$$

$$+ \frac{C_i}{\lambda} \left\{ \frac{\sin [\pi (1 - \omega)d]}{(1 - \omega)} + \frac{\sin [\pi (1 + \omega)d]}{(1 + \omega)} \right\}$$

where $d = \Delta/\lambda$. The integral for G_2 (the coefficient of F_2') is the same with $\bar{r}_i$ and $\hat{i}$ reflected in the ground plane. The terms G_1 and G_2 and other necessary quantities are passed to subroutine GWAVE through COMMON/GWAV/. GWAVE returns the field components

E_ρ^v = ρ' component of field due to vertical current component

E_z^v = z component of field due to vertical current component

E_ρ^h = ρ' component of field due to horizontal current component

E_ϕ^h = ϕ' component of field due to horizontal current component

E_z^h = z component of field due to horizontal current component

The common factor exp(-jkR) occurring in F_1' and F_2' is omitted from the field components and included in the total field after summation.

These field components are then combined to form the total field in x, y, z components and summed for each segment. The field is finally converted to r, θ, ϕ components in a spherical coordinate system coinciding with the x, y, z coordinate system.

The approximations involved in the calculation of the surface wave are valid to second order in u^2, where

$u = k/k_2$

k = wave number in free space

k_2 = wave number in ground medium

The approximations are valid for practical ground parameters. To ensure that the expressions are not used in an invalid range, however, the surface wave is not computed if $|u|$ is greater than 0.5. Rather, subroutine FFLD is called, and the resulting space wave is multiplied by the range factor exp(-jkR)/(R/λ). The radial field component will be zero in this case. FFLD is also called if R/λ is greater than 10^5, or if there is no ground present.

SYMBOL DICTIONARY

A = coefficient of A_i/λ in $2\pi G_1$ and $2\pi G_2$

ABS = external routine (absolute value)

ARG = argument of exp( ) for phase factor

ATAN = external routine (arctangent)

B = coefficient of B_i/λ in $2\pi G_1$ and $2\pi G_2$

BOO = sin (BOT)/BOT

BOT = $\pi(1 - \omega)d$

C = coefficient of C_i/λ in $2\pi G_1$ and $2\pi G_2$

CAB(I) = $\cos\alpha \cos\beta$ for segment I

CABS = external routine (magnitude of complex number)

CALP = $\cos\alpha$

CBET = $\cos\beta$

CIX, CIY, CIZ = x, y, z components in summation for field

CMPLX = external routine (forms complex number)

COS = external routine (cosine)

CPH = $\cos\phi'$

DX, DY, DZ = x, y, z components of $\hat{i}$

EL = πd

EPH = E^h_ϕ or $E^h_\phi \cos\alpha$ (ϕ' component of total field of segment i)

EPI = ϕ component of field of structure

ERD = R component of field of structure

ERH = E^h_ρ and ρ' component of total field of segment i

ERV = E^v_ρ

ETH = θ component of field of structure

EX = x component of field for segment i

EXA = phase factor at GD30 and GD130:

$G_1 \exp(jk\hat{R}_1 \cdot \overline{r}_i)$ or $G_2 \exp(jk\hat{R}_2 \cdot \overline{r}'_i)$ at GD109

EY = y component of field for segment i

EZH = E^h_z and z component of total field of segment i

EZV = E^v_z

FFLD = external routine (computes space wave)

GWAVE = external routine (computes E_ρ^v, E_ρ^h ,...)

I = DO loop index (i)

K = DO loop index (loop over segment and image)

KSYMP = 1 if ground is present; 0 otherwise

OMEGA = ω

PHI = ϕ

PHX = x component of $\hat{\phi}$

PHY = y component of $\hat{\phi}$

PI = π

R = R/λ

RFL = sign factor to reflect segment coordinates in ground

RHO = ρ/λ

RHP = ρ'/λ

RHS = $(\rho'/\lambda)^2$

RHX = x component of $\hat{\rho}'$

RHY = y component of $\hat{\rho}'$

RI = imaginary part of $2\pi G_1$ or $2\pi G_2$

RIX = x component of $\bar{R}_1/\lambda$ or $\bar{R}_2/\lambda$

RIY = y component of $\bar{R}_1/\lambda$ or $\bar{R}_2/\lambda$

RIZ = z component of $\bar{R}_1/\lambda$ or $\bar{R}_2/\lambda$

RNX, RNY, RNZ = x, y, z components of $\hat{R}_1$ or $\hat{R}_2$ or $\hat{R}$

RR = real part of $2\pi G_1$ or $2\pi G_2$

RX = x component of $\bar{\rho}/\lambda$

RXYZ = R_1/λ or R_2/λ (for s = 0)

RY = y component of $\bar{\rho}/\lambda$

RZ = z/λ

SAB(I) = cos α sin β

SBET = sin β

SILL = $\pi d\omega$

SIN = external routine (sine)

SPH = sin ϕ'

SQRT	= external routine (square root)
THET	= θ in spherical coordinate system
THX	= x component of $\hat{\theta}$
THY	= y component of $\hat{\theta}$
THZ	= z component of $\hat{\theta}$
TOO	= sin (TOP)/TOP
TOP	= $\pi(1 + \omega)d$
TP	= 2π
U	= u
UX	= u
U2	= u^2
XX1	= $G_1 \exp(jk\hat{R}_1 \cdot \overline{r}_i)$
XX2	= $G_2 \exp(jk\hat{R}_2 \cdot \overline{r}'_i)$

CONSTANTS

1.E-20	= tolerance in test for zero
1.E-7	= tolerance in test for zero
1.E-6	= tolerance in test for zero
0.5	= upper limit for $\lvert u \rvert$
3.141592654	= π
6.283185308	= 2π
1.E+5	= upper limit for R/λ

```
      SUBROUTINE GFLD (RHO,PHI,RZ,ETH,EPI,ERD,UX,KSYMP)
C
C     GFLD COMPUTES THE RADIATED FIELD INCLUDING GROUND WAVE.
C
      COMPLEX CUR,EPI,CIX,CIY,CIZ,EXA,XX1,XX2,U,U2,ERV,EZV,ERH,EPH
      COMPLEX EZH,EX,EY,ETH,UX,ERD
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),CII(300
     1),CUR(900)
      COMMON /GWAV/ U,U2,XX1,XX2,R1,R2,ZMH,ZPH
      DIMENSION CAB(1), SAB(1)
      EQUIVALENCE (CAB(1),ALP(1)), (SAB(1),BET(1))
      DATA PI,TP/3.141592654,6.283185308/
      R=SQRT(RHO*RHO+RZ*RZ)
      IF (KSYMP.EQ.1) GO TO 1
      IF (CABS(UX).GT..5) GO TO 1
      IF (R.GT.1.E5) GO TO 1
      GO TO 4
C
C     COMPUTATION OF SPACE WAVE ONLY
C
1     IF (RZ.LT.1.E-20) GO TO 2
      THET=ATAN(RHO/RZ)
      GO TO 3
2     THET=PI*.5
3     CALL FFLD (THET,PHI,ETH,EPI)
      ARG=-TP*R
      EXA=CMPLX(COS(ARG),SIN(ARG))/R
      ETH=ETH*EXA
      EPI=EPI*EXA
      ERD=(0.,0.)
      RETURN
C
C     COMPUTATION OF SPACE AND GROUND WAVES.
C
4     U=UX
      U2=U*U
      PHX=-SIN(PHI)
      PHY=COS(PHI)
      RX=RHO*PHY
      RY=-RHO*PHX
      CIX=(0.,0.)
      CIY=(0.,0.)
      CIZ=(0.,0.)
C
C     SUMMATION OF FIELD FROM INDIVIDUAL SEGMENTS
C
      DO 17 I=1,N
      DX=CAB(I)
      DY=SAB(I)
      DZ=SALP(I)
      RIX=RX-X(I)
      RIY=RY-Y(I)
      RHS=RIX*RIX+RIY*RIY
      RHP=SQRT(RHS)
      IF (RHP.LT.1.E-6) GO TO 5
      RHX=RIX/RHP
      RHY=RIY/RHP
      GO TO 6
5     RHX=1.
      RHY=0.
```

```
6     CALP=1.-DZ*DZ
      IF (CALP.LT.1.E-6) GO TO 7
      CALP=SQRT(CALP)
      CBET=DX/CALP
      SBET=DY/CALP
      CPH=RHX*CBET+RHY*SBET
      SPH=RHY*CBET-RHX*SBET
      GO TO 8
7     CPH=RHX
      SPH=RHY
8     EL=PI*SI(I)
      RFL=-1.
C
C     INTEGRATION OF (CURRENT)*(PHASE FACTOR) OVER SEGMENT AND IMAGE FOR
C     CONSTANT, SINE, AND COSINE CURRENT DISTRIBUTIONS
C
      DO 16 K=1,2
      RFL=-RFL
      RIZ=RZ-Z(I)*RFL
      RXYZ=SQRT(RIX*RIX+RIY*RIY+RIZ*RIZ)
      RNX=RIX/RXYZ
      RNY=RIY/RXYZ
      RNZ=RIZ/RXYZ
      OMEGA=-(RNX*DX+RNY*DY+RNZ*DZ*RFL)
      SILL=OMEGA*EL
      TOP=EL+SILL
      BOT=EL-SILL
      IF (ABS(OMEGA).LT.1.E-7) GO TO 9
      A=2.*SIN(SILL)/OMEGA
      GO TO 10
9     A=(2.-OMEGA*OMEGA*EL*EL/3.)*EL
10    IF (ABS(TOP).LT.1.E-7) GO TO 11
      TOO=SIN(TOP)/TOP
      GO TO 12
11    TOO=1.-TOP*TOP/6.
12    IF (ABS(BOT).LT.1.E-7) GO TO 13
      BOO=SIN(BOT)/BOT
      GO TO 14
13    BOO=1.-BOT*BOT/6.
14    B=EL*(BOO-TOO)
      C=EL*(BOO+TOO)
      RR=A*AIR(I)+B*BII(I)+C*CIR(I)
      RI=A*AII(I)-B*BIR(I)+C*CII(I)
      ARG=TP*(X(I)*RNX+Y(I)*RNY+Z(I)*RNZ*RFL)
      EXA=CMPLX(COS(ARG),SIN(ARG))*CMPLX(RR,RI)/TP
      IF (K.EQ.2) GO TO 15
      XX1=EXA
      R1=RXYZ
      ZMH=RIZ
      GO TO 16
15    XX2=EXA
      R2=RXYZ
      ZPH=RIZ
16    CONTINUE
C
C     CALL SUBROUTINE TO COMPUTE THE FIELD OF SEGMENT INCLUDING GROUND
C     WAVE.
C
      CALL GWAVE (ERV,EZV,ERH,EZH,EPH)
      ERH=ERH*CPH*CALP+ERV*DZ
      EPH=EPH*SPH*CALP
      EZH=EZH*CPH*CALP+EZV*DZ
      EX=ERH*RHX-EPH*RHY
      EY=ERH*RHY+EPH*RHX
```

```
         CIX=CIX+EX                                            GD 129
         CIY=CIY+EY                                            GD 130
   17    CIZ=CIZ+EZH                                           GD 131
         ARG=-TP*R                                             GD 132
         EXA=CMPLX(COS(ARG),SIN(ARG))                          GD 133
         CIX=CIX*EXA                                           GD 134
         CIY=CIY*EXA                                           GD 135
         CIZ=CIZ*EXA                                           GD 136
         RNX=RX/R                                              GD 137
         RNY=RY/R                                              GD 138
         RNZ=RZ/R                                              GD 139
         THX=RNZ*PHY                                           GD 140
         THY=-RNZ*PHX                                          GD 141
         THZ=-RHO/R                                            GD 142
         ETH=CIX*THX+CIY*THY+CIZ*THZ                           GD 143
         EPI=CIX*PHX+CIY*PHY                                   GD 144
         ERD=CIX*RNX+CIY*RNY+CIZ*RNZ                           GD 145
         RETURN                                                GD 146
         END                                                   GD 147-
```

GFOUT

PURPOSE

To write the NGF file.

METHOD

The contents of the COMMON blocks in GFOUT are written to file 20. If ICASE is 3 or 5 the blocks of the LU decomposition of matrix A are on file 13 in ascending order and on file 14 in descending order. Both files are written to file 20.

SYMBOL DICTIONARY

IGFL = NGF file number

IOUT = number of elements in matrix

NEQ = order of matrix A

NOP = number of symmetric sections

NPEQ = number of unknowns for a symmetric section

GFOUT

```
      SUBROUTINE GFOUT
C
C     WRITE N.G.F. FILE
C
      COMPLEX CM,SSX,ZRATI,ZRATI2,T1,ZARRAY,AR1,AR2,AR3,EPSCF,FRATI
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /CMB/ CM(4000)
      COMMON /ANGL/ SALP(300)
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      COMMON /GGRID/ AR1(11,10,4),AR2(17,5,4),AR3(9,8,4),EPSCF,DXA(3),DY
     1A(3),XSA(3),YSA(3),NXA(3),NYA(3)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      COMMON /SMAT/ SSX(16,16)
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF
      COMMON /SAVE/ IP(600),KCOM,COM(13,5),EPSR,SIG,SCRWLT,SCRWRT,FMHZ
      DATA IGFL/20/
      NEQ=N+2*M
      NPEQ=NP+2*MP
      NOP=NEQ/NPEQ
      WRITE (IGFL) N,NP,M,MP,WLAM,FMHZ,IPSYM,KSYMP,IPERF,NRADL,EPSR,SIG,
     1SCRWLT,SCRWRT,NLOAD,KCOM
      IF (N.EQ.0) GO TO 1
      WRITE (IGFL) (X(I),I=1,N),(Y(I),I=1,N),(Z(I),I=1,N)
      WRITE (IGFL) (SI(I),I=1,N),(BI(I),I=1,N),(ALP(I),I=1,N)
      WRITE (IGFL) (BET(I),I=1,N),(SALP(I),I=1,N)
      WRITE (IGFL) (ICON1(I),I=1,N),(ICON2(I),I=1,N)
      WRITE (IGFL) (ITAG(I),I=1,N)
      IF (NLOAD.GT.0) WRITE (IGFL) (ZARRAY(I),I=1,N)
1     IF (M.EQ.0) GO TO 2
      J=LD-M+1
      WRITE (IGFL) (X(I),I=J,LD),(Y(I),I=J,LD),(Z(I),I=J,LD)
      WRITE (IGFL) (SI(I),I=J,LD),(BI(I),I=J,LD),(ALP(I),I=J,LD)
      WRITE (IGFL) (BET(I),I=J,LD),(SALP(I),I=J,LD)
      WRITE (IGFL) (ICON1(I),I=J,LD),(ICON2(I),I=J,LD)
      WRITE (IGFL) (ITAG(I),I=J,LD)
2     WRITE (IGFL) ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT
      IF (IPERF.EQ.2) WRITE (IGFL) AR1,AR2,AR3,EPSCF,DXA,DYA,XSA,YSA,NXA
     1,NYA
      IF (NOP.GT.1) WRITE (IGFL) ((SSX(I,J),I=1,NOP),J=1,NOP)
      WRITE (IGFL) (IP(I),I=1,NEQ),COM
      IF (ICASE.GT.2) GO TO 3
      IOUT=NEQ*NPEQ
      WRITE (IGFL) (CM(I),I=1,IOUT)
      GO TO 12
3     IF (ICASE.NE.4) GO TO 5
      REWIND 13
      I=NPEQ*NPEQ
      DO 4 K=1,NOP
      READ (13) (CM(J),J=1,I)
4     WRITE (IGFL) (CM(J),J=1,I)
      REWIND 13
      GO TO 12
5     REWIND 13
      REWIND 14
      IF (ICASE.EQ.5) GO TO 8
      IOUT=NPBLK*NEQ*2
      DO 6 I=1,NBLOKS
      CALL BLCKIN (CM,13,1,IOUT,1,201)
6     CALL BLCKOT (CM,IGFL,1,IOUT,1,202)
      DO 7 I=1,NBLOKS
```

```
      CALL BLCKIN (CM,14,1,IOUT,1,203)
    7 CALL BLCKOT (CM,IGFL,1,IOUT,1,204)
      GO TO 12
    8 IOUT=NPSYM*NPEQ*2
      DO 11 IOP=1,NOP
      DO 9 I=1,NBLSYM
      CALL BLCKIN (CM,13,1,IOUT,1,205)
    9 CALL BLCKOT (CM,IGFL,1,IOUT,1,206)
      DO 10 I=1,NBLSYM
      CALL BLCKIN (CM,14,1,IOUT,1,207)
   10 CALL BLCKOT (CM,IGFL,1,IOUT,1,208)
   11 CONTINUE
      REWIND 13
      REWIND 14
   12 REWIND IGFL
      PRINT 13, IGFL,IMAT
      RETURN
C
   13 FORMAT (///,44H ****NUMERICAL GREEN'S FUNCTION FILE ON TAPE,I3,5H
     1****,/,5X,16HMATRIX STORAGE -,I7,16H COMPLEX NUMBERS,///)
      END
```

GH

PURPOSE

To compute the function that is numerically integrated for the near H field of a segment.

METHOD

The value returned by GH is

$$G = \left[\frac{1}{(kr)^3} + \frac{j}{(kr)^2} \right] \exp(-jkr) \ ,$$

where

$r = \left[\rho'^2 + (z - z')^2 \right]^{1/2}$

ρ' = ρ coordinate of the field observation point in a cylindrical coordinate system with origin at the center of the source segment and z axis oriented along the source segment

z' = z coordinate of the field observation point in the cylindrical coordinate system

z = z coordinate of the integration point on the source segment

$k = 2\pi/\lambda$

SYMBOL DICTIONARY

CKR = cos kr

HR = real part of G

HI = imaginary part of G

R = kr

RHKS = $(k\rho')^2$

RR2 = $1/(kr)^2$

RR3 = $1/(kr)^3$

RS = $(kr)^2$

SKR = sin kr

ZK = kz

ZPK = kz'

```
      SUBROUTINE GH (ZK,HR,HI)
C     INTEGRAND FOR H FIELD OF A WIRE
      COMMON /TMH/ ZPK,RHKS
      RS=ZK-ZPK
      RS=RHKS+RS*RS
      R=SQRT(RS)
      CKR=COS(R)
      SKR=SIN(R)
      RR2=1./RS
      RR3=RR2/R
      HR=SKR*RR2+CKR*RR3
      HI=CKR*RR2-SKR*RR3
      RETURN
      END
```

GWAVE

PURPOSE

To compute the components of electric field due to an electric current element over a ground plane at intermediate distances, including the surface wave field.

METHOD

Approximate expressions for the electric field of a vertical or horizontal infinitesimal current element above a ground plane, including surface wave, were derived by K. A. Norton (ref. 2). The geometry is shown in figure 6a for a current element at height a above the ground plane and field observation point at p. The current element is located on the z' axis, and the horizontal current element is directed along the x' axis. The vertical current element produces z' and ρ' field components given by

$$E_z^v = -\frac{j\eta I d\ell}{2\lambda}\left\{\cos^2\psi'\,\frac{\exp(-jkR_1)}{R_1} + R_v\cos^2\psi\,\frac{\exp(-jkR_2)}{R_2}\right.$$

$$+ (1-R_v)\cos^2\psi\; F\,\frac{\exp(-jkR_2)}{R_2}$$

$$+ u\sqrt{1-u^2\cos^2\psi}\,\sin\psi\; 2\,\frac{\exp(-jkR_2)}{jkR_2^2}$$

$$+ \frac{\exp(-jkR_1)}{R_1}\left(\frac{1}{jkR_1}+\frac{1}{(jkR_1)^2}\right)(1-3\sin^2\psi')$$

$$\left.+ \frac{\exp(-jkR_2)}{R_2}\left(\frac{1}{jkR_2}+\frac{1}{(jkR_2)^2}\right)(1-3\sin^2\psi)\right\},$$

$$E_\rho^v = \frac{j\eta I d\ell}{2\lambda}\left\{\sin\psi'\cos\psi'\frac{\exp(-jkR_1)}{R_1} + R_v\sin\psi\cos\psi\frac{\exp(-jkR_2)}{R_2}\right.$$

$$- \cos\psi(1 - R_v)\, u\sqrt{1 - u^2\cos^2\psi}\; F\,\frac{\exp(-jkR_2)}{R_2}$$

$$- \sin\psi\cos\psi\,(1 - R_v)\,\frac{\exp(-jkR_2)}{jkR_2^2}$$

$$+ 3\sin\psi'\cos\psi'\left(\frac{1}{jkR_1} + \frac{1}{(jkR_1)^2}\right)\frac{\exp(-jkR_1)}{R_1}$$

$$- \cos\psi\, u\sqrt{1 - u^2\cos^2\psi}\,(1 - R_v)\,\frac{\exp(-jkR_2)}{2jkR_2^2}$$

$$\left. + 3\sin\psi\cos\psi\left(\frac{1}{jkR_2} + \frac{1}{(jkR_2)^2}\right)\frac{\exp(-jkR_2)}{R_2}\right\},$$

where

$F = 1 - j\sqrt{\pi w}\,\exp(-w)\,\mathrm{erfc}\,(j\sqrt{w})$

$\mathrm{erfc}(z) = 1 - \mathrm{erf}(z)$

$\mathrm{erf}(z) = 2/\sqrt{\pi}\int_o^z \exp(-t^2)\,dt$ (error function)

$w = 4p_1/(1 - R_v)^2$

$p_1 = -jkR_2u^2\,(1 - u^2\cos^2\psi)/(2\cos^2\psi)$

$R_v = \dfrac{\sin\psi - u\sqrt{1 - u^2\cos^2\psi}}{\sin\psi + u\sqrt{1 - u^2\cos^2\psi}}$

$u = k/k_2$

k = wave number in free space

k_2 = wave number in lower medium

$\sin\psi = (z + a)/R_2$

$\sin\psi' = (z - a)/R_1$

The horizontal current element directed along the x' axis produces ρ', ϕ', and z' field components given by

$$E_z^h = \frac{j\eta I d\ell}{2\lambda} \cos \phi' \Bigg\{ \sin \psi' \cos \psi' \frac{\exp(-jkR_1)}{R_1}$$

$$- R_v \sin \psi \cos \psi \cdot \frac{\exp(-jkR_2)}{R_2}$$

$$+ \cos \psi (1 - R_v) u\sqrt{1 - u^2 \cos^2 \psi}\, F \frac{\exp(-jkR_2)}{R_2}$$

$$+ \sin \psi \cos \psi (1 - R_v) \frac{\exp(-jkR_2)}{jkR_2^2}$$

$$+ 3 \sin \psi' \cos \psi' \left(\frac{1}{jkR_1} + \frac{1}{(jkR_1)^2}\right) \frac{\exp(-jkR_1)}{R_1}$$

$$+ \cos \psi (1 - R_v) u\sqrt{1 - u^2 \cos^2 \psi} \frac{\exp(-jkR_2)}{2jkR_2^2}$$

$$- 3 \sin \psi \cos \psi \left(\frac{1}{jkR_2} + \frac{1}{(jkR_2)^2}\right) \frac{\exp(-jkR_2)}{R_2} \Bigg\} ,$$

$$E^h_\rho = \frac{-j\eta I d\ell}{2\lambda} \cos\phi' \Bigg\{ \sin^2\psi' \frac{\exp(-jkR_1)}{R_1} - R_v \sin^2\psi \frac{\exp(-jkR_2)}{R_2}$$

$$- (1 - u^2\cos^2\psi)\, u^2\, (1 - R_v)\, F \frac{\exp(-jkR_2)}{R_2}$$

$$+ \left(\frac{1}{jkR_1} + \frac{1}{(jkR_1)^2}\right) (1 - 3\cos^2\psi') \frac{\exp(-jkR_1)}{R_1}$$

$$- \left(\frac{1}{jkR_2} + \frac{1}{(jkR_2)^2}\right) (1 - 3\cos^2\psi) \left[1 - u^2\, (1 + R_v) - u^2 (1 - R_v) F\right]$$

$$\times \frac{\exp(-jkR_2)}{R_2} + u^2 \cos^2\psi\, (1 - R_v) \left(1 + \frac{1}{jkR_2}\right)$$

$$\times \left[F\left(u^2 (1 - u^2\cos^2\psi) - \sin^2\psi + \frac{1}{jkR_2}\right) - \frac{1}{jkR_2}\right] \frac{\exp(-jkR_2)}{R_2} \Bigg\},$$

$$E^h_\phi = \frac{j\eta I d\ell}{2\lambda} \sin\phi' \Bigg\{ \frac{\exp(-jkR_1)}{R_1} - R_h \frac{\exp(-jkR_2)}{R_2}$$

$$+ (R_h + 1)\, G \frac{\exp(-jkR_2)}{R_2} + \left(1 + \frac{1}{jkR_1}\right) \frac{\exp(-jkR_1)}{jkR_1^2}$$

$$- \left(1 + \frac{1}{jkR_2}\right) [1 - u^2\, (1 + R_v) - u^2\, (1 - R_v) F]\, \frac{\exp(-jkR_2)}{jkR_2^2}$$

$$- \frac{u^2 (1 - R_v)}{2} \left[F\left(u^2\, (1 - u^2\cos^2\psi) - \sin^2\psi + \frac{1}{jkR_2}\right) - \frac{1}{jkR_2}\right]$$

$$\times \frac{\exp(-jkR_2)}{jkR_2^2} \Bigg\},$$

where

$$G = [1 - j\sqrt{\pi v}\exp(-v)\,\mathrm{erfc}\,(j\sqrt{v})],$$

$$v = 4q_1/\,(1 + R_h)^2$$

$$q_1 = -jkR_2\,(1 - u^2\cos^2\psi)/(2u^2\cos^2\psi)$$

$$R_h = \frac{\sqrt{1 - u^2\cos^2\psi} - u\sin\psi}{\sqrt{1 - u^2\cos^2\psi} + u\sin\psi}$$

The approximations in these expressions are valid for R_1 and R_2 greater than about a wavelength and to second order in u^2. In each equation, the first term represents the direct space wave field of the current element, the second term is the space wave field reflected from the ground, and the following higher order terms involving F and G represent the ground wave. It may be noted that the coefficients R_v and R_h are the Fresnel reflection coefficients for vertical and horizontal polarization, respectively.

To obtain the field due to a structure, these expressions are integrated over each segment and the fields of the segments are summed in subroutine GFLD. For integration, R_1 and R_2 are the distances from the integration point ℓ on the segment to point p. Since R_1 and R_2 are assumed large compared to the segment length, R_1, R_2, ψ, and ψ' are considered constant during integration over the segment except where jkR_1 and jkR_2 occur in exponential functions. Thus, if s represents distance along the segment, the integral of each expression over the segment is obtained by replacing $(Id\ell/\lambda^2)\exp(-jkR_1)$ and $(Id\ell/\lambda^2)\exp(-jkR_2)$ by XX1 and XX2 from subroutine GFLD. A factor of exp(-jkR) is omitted from the fields and is included after summation in GFLD. Including a factor of $1/\lambda^2$ in XX1 and XX2 makes a factor of λ available to normalize R_1 and R_2 in the denominators of the field expressions. The factors sin ϕ' or cos ϕ' are omitted from the fields due to a horizontal current element in GWAVE and are supplied later.

SYMBOL DICTIONARY

CPP = $\cos\psi$

CPPP = $\cos\psi'$

CPPP2 = $\cos^2\psi'$

CPP2 = $\cos^2\psi$

ECON = $-j\eta/2$ (η = impedance of free space)

EPH = $E^h_\phi/\sin\phi'$

ERH = $E^h_\rho/\cos\phi'$

ERV = E^v_ρ

EZH = $E^h_z/\cos\phi'$

EZV = E^v_z

F = F

FJ = $j = \sqrt{-1}$

G = G

OMR = $1 - R_v$

PI = π

P1 = p_1

Q1 = q_1

RH = R_h

RK1 = $-jkR_1$

RK2 = $-jkR_2$

RV = R_v

R1 = R_1/λ

R2 = R_2/λ

SPP = $\sin\psi$

SPPP = $\sin\psi'$

SPPP2 = $\sin^2\psi'$

SPP2 = $\sin^2\psi$

TPJ = $2\pi j$

T1 = $1 - u^2\cos^2\psi$

T2 = $\sqrt{T1}$

T3 = $-[1/(jkR_1) + 1/(jkR_1)^2]$

GWAVE

T4	$= -[1/(jkR_2) + 1/(jkR_2)^2]$
U	$= u$
U2	$= u^2$
V	$= v$
W	$= w$
XR1	$= XX1/(R/\lambda)$
XR2	$= XX2/(R/\lambda)$
XX1	$= G_1 \exp(jk\hat{R}_1 \cdot \bar{r}_i)$
XX2	$= G_2 \exp(jk\hat{R}_2 \cdot \bar{r}_i')$

$$\left.\begin{array}{l} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \end{array}\right\} = \text{first, second, ..., seventh term in each field expression}$$

ZMH	$= z - a$
ZPH	$= z + a$

CONSTANTS

(0., 1.)	$= j = \sqrt{-1}$
(0., 6.2831853)	$= 2\pi j$
(0., -188.363)	$= -j\eta/2$
3.1415926	$= \pi$

```
      SUBROUTINE GWAVE (ERV,EZV,ERH,EZH,EPH)
C
C     GWAVE COMPUTES THE ELECTRIC FIELD, INCLUDING GROUND WAVE, OF A
C     CURRENT ELEMENT OVER A GROUND PLANE USING FORMULAS OF K.A. NORTON
C     (PROC. IRE, SEPT., 1937, PP.1203,1236.)
C
      COMPLEX FJ,TPJ,U2,U,RK1,RK2,T1,T2,T3,T4,P1,RV,OMR,W,F,Q1,RH,V,G,XR
     11,XR2,X1,X2,X3,X4,X5,X6,X7,EZV,ERV,EZH,ERH,EPH,XX1,XX2,ECON,FBAR
      COMMON /GWAV/ U,U2,XX1,XX2,R1,R2,ZMH,ZPH
      DIMENSION FJX(2), TPJX(2), ECONX(2)
      EQUIVALENCE (FJ,FJX), (TPJ,TPJX), (ECON,ECONX)
      DATA PI/3.141592654/,FJX/0.,1./,TPJX/0.,6.283185308/
      DATA ECONX/0.,-188.367/
      SPPP=ZMH/R1
      SPPP2=SPPP*SPPP
      CPPP2=1.-SPPP2
      IF (CPPP2.LT.1.E-20) CPPP2=1.E-20
      CPPP=SQRT(CPPP2)
      SPP=ZPH/R2
      SPP2=SPP*SPP
      CPP2=1.-SPP2
      IF (CPP2.LT.1.E-20) CPP2=1.E-20
      CPP=SQRT(CPP2)
      RK1=-TPJ*R1
      RK2=-TPJ*R2
      T1=1.-U2*CPP2
      T2=CSQRT(T1)
      T3=(1.-1./RK1)/RK1
      T4=(1.-1./RK2)/RK2
      P1=RK2*U2*T1/(2.*CPP2)
      RV=(SPP-U*T2)/(SPP+U*T2)
      OMR=1.-RV
      W=1./OMR
      W=(4.,0.)*P1*W*W
      F=FBAR(W)
      Q1=RK2*T1/(2.*U2*CPP2)
      RH=(T2-U*SPP)/(T2+U*SPP)
      V=1./(1.+RH)
      V=(4.,0.)*Q1*V*V
      G=FBAR(V)
      XR1=XX1/R1
      XR2=XX2/R2
      X1=CPPP2*XR1
      X2=RV*CPP2*XR2
      X3=OMR*CPP2*F*XR2
      X4=U*T2*SPP*2.*XR2/RK2
      X5=XR1*T3*(1.-3.*SPPP2)
      X6=XR2*T4*(1.-3.*SPP2)
      EZV=(X1+X2+X3-X4-X5-X6)*ECON
      X1=SPPP*CPPP*XR1
      X2=RV*SPP*CPP*XR2
      X3=CPP*OMR*U*T2*F*XR2
      X4=SPP*CPP*OMR*XR2/RK2
      X5=3.*SPPP*CPPP*T3*XR1
      X6=CPP*U*T2*OMR*XR2/RK2*.5
      X7=3.*SPP*CPP*T4*XR2
      ERV=-(X1+X2-X3+X4-X5+X6-X7)*ECON
      EZH=-(X1-X2+X3-X4-X5-X6+X7)*ECON
      X1=SPPP2*XR1
      X2=RV*SPP2*XR2
      X4=U2*T1*OMR*F*XR2
      X5=T3*(1.-3.*CPPP2)*XR1
      X6=T4*(1.-3.*CPP2)*(1.-U2*(1.+RV)-U2*OMR*F)*XR2
      X7=U2*CPP2*OMR*(1.-1./RK2)*(F*(U2*T1-SPP2-1./RK2)+1./RK2)*XR2
```

```
      ERH=(X1-X2-X4-X5+X6+X7)*ECON
      X1=XR1
      X2=RH*XR2
      X3=(RH+1.)*G*XR2
      X4=T3*XR1
      X5=T4*(1.-U2*(1.+RV)-U2*OMR*F)*XR2
      X6=.5*U2*OMR*(F*(U2*T1-SPP2-1./RK2)+1./RK2)*XR2/RK2
      EPH=-(X1-X2+X3-X4+X5+X6)*ECON
      RETURN
      END
```

GX

PURPOSE

To evaluate terms for the field contribution due to segment ends in the thin wire kernel.

SYMBOL DICTIONARY

GZ = $\exp(-jkr)/r = G_0$

GZP = $-(1 + jkr)\exp(-jkr)/r^3$

R = r

R2 = $r^2 = \rho^2 + z^2$

RH = ρ

RK = kR

XK = $2\pi/\lambda$

ZZ = z

CODE LISTING

```
      SUBROUTINE GX (ZZ,RH,XK,GZ,GZP)                              GX   1
C     SEGMENT END CONTRIBUTIONS FOR THIN WIRE APPROX.              GX   2
      COMPLEX GZ,GZP                                               GX   3
      R2=ZZ*ZZ+RH*RH                                               GX   4
      R=SQRT(R2)                                                   GX   5
      RK=XK*R                                                      GX   6
      GZ=CMPLX(COS(RK),-SIN(RK))/R                                 GX   7
      GZP=-CMPLX(1.,RK)*GZ/R2                                      GX   8
      RETURN                                                       GX   9
      END                                                          GX  10-
```

GXX

PURPOSE

To evaluate terms for the field contribution due to segment ends in the extended thin wire kernel.

METHOD

Equations 89 through 94 in Part I are evaluated for $\rho > a$, and equations 99 through 103 for $\rho < a$. Several variables are used for storage of intermediate results before being set to their final values.

SYMBOL DICTIONARY

A = radius of source segment, a

A2 = a^2

C1 = $1 + jkr_0$

C2 = $3(1 + jkr_0) - k^2r_0^2$

C3 = $(6 + jkr_0)k^2r_0^2 - 15(1 + jkr_0)$

G1 = G_1

G1P = $\partial G_1/\partial z'$

G2 = G_2

G2P = $\partial G_2/\partial z'$

G3 = $\partial G_1/\partial \rho$

GZ = G_0

GZP = $\partial G_0/\partial z'$

IRA = 1 to indicate $\rho < a$

R = r_0

R2 = r_0^2

R4 = r_0^4

RH = ρ

RH2 = ρ^2

RK = kr_0

RK2 = $k^2r_0^2$

T1 = $a^2\rho^2/4r^4$

T2 = $a^2/2r^2$

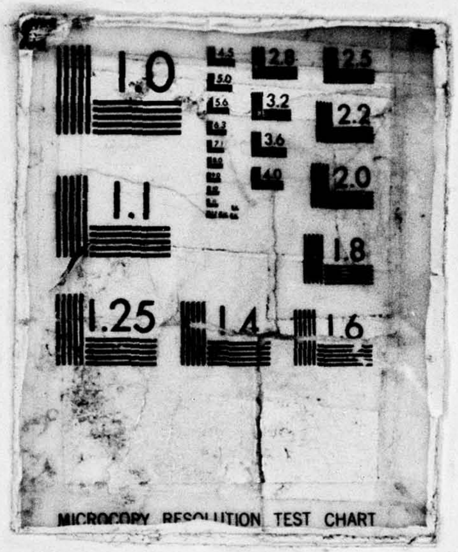
XK = $k = 2\pi/\lambda$

ZZ = $z' - z$

```
      SUBROUTINE GXX (ZZ,RH,A,A2,XK,IRA,G1,G1P,G2,G2P,G3,GZP)            GY   1
C     SEGMENT END CONTRIBUTIONS FOR EXT. THIN WIRE APPROX.               GY   2
      COMPLEX GZ,C1,C2,C3,G1,G1P,G2,G2P,G3,GZP                           GY   3
      R2=ZZ*ZZ+RH*RH                                                     GY   4
      R=SQRT(R2)                                                         GY   5
      R4=R2*R2                                                           GY   6
      RK=XK*R                                                            GY   7
      RK2=RK*RK                                                          GY   8
      RH2=RH*RH                                                          GY   9
      T1=.25*A2*RH2/R4                                                   GY  10
      T2=.5*A2/R2                                                        GY  11
      C1=CMPLX(1.,RK)                                                    GY  12
      C2=3.*C1-RK2                                                       GY  13
      C3=CMPLX(6.,RK)*RK2-15.*C1                                         GY  14
      GZ=CMPLX(COS(RK),-SIN(RK))/R                                       GY  15
      G2=GZ*(1.+T1*C2)                                                   GY  16
      G1=G2-T2*C1*GZ                                                     GY  17
      GZ=GZ/R2                                                           GY  18
      G2P=GZ*(T1*C3-C1)                                                  GY  19
      GZP=T2*C2*GZ                                                       GY  20
      G3=G2P+GZP                                                         GY  21
      G1P=G3*ZZ                                                          GY  22
      IF (IRA.EQ.1) GO TO 2                                              GY  23
      G3=(G3+GZP)*RH                                                     GY  24
      GZP=-ZZ*C1*GZ                                                      GY  25
      IF (RH.GT.1.E-10) GO TO 1                                          GY  26
      G2=0.                                                              GY  27
      G2P=0.                                                             GY  28
      RETURN                                                             GY  29
1     G2=G2/RH                                                           GY  30
      G2P=G2P*ZZ/RH                                                      GY  31
      RETURN                                                             GY  32
2     T2=.5*A                                                            GY  33
      G2=-T2*C1*GZ                                                       GY  34
      G2P=T2*GZ*C2/R2                                                    GY  35
      G3=RH2*G2P-A*GZ*C1                                                 GY  36
      G2P=G2P*ZZ                                                         GY  37
      GZP=-ZZ*C1*GZ                                                      GY  38
      RETURN                                                             GY  39
      END                                                                GY  40-
```

AD-A075 289 NUMERICAL ELECTROMAGNETIC CODE (NEC)-METHOD OF MOMENTS A USER-ORIENTED CO..(U) NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA G J BURKE ET AL. 18 JUL 77 NOSC/TD-116-VOL-1 4/6

UNCLASSIFIED AFWL-TR-76-320-VOL-1 F/G 12/5 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

HFK

PURPOSE

To compute the near H field of a uniform current filament by numerical integration.

METHOD

The H field of a current filament of length Δ with uniform current distribution of magnitude I = λ is

$$H_\phi = \frac{k\rho'}{2} \int_{-k\Delta/2}^{k\Delta/2} \left[\frac{1}{(kr)^3} + \frac{j}{(kr)^2} \right] \exp(-jkr)\, d(kz) ,$$

where r, ρ', z' and z are defined in the description of subroutine GH. The numerical integration is performed by the method of Romberg quadrature with variable interval width, which is described in the discussion of subroutine INTX. The integral is multiplied by kρ'/2 at HF79 and HF80 in the code.

SYMBOL DICTIONARY

This listing excludes those variables used in the numerical quadrature algorithm, which are defined under subroutine INTX.

RHK = kρ'

RHKS = $(k\rho')^2$

SGI = imaginary part of H_ϕ

SGR = real part of H_ϕ

ZPK = kz' (z' = z coordinate of observation point)

ZPKX = ZPK

```
      SUBROUTINE HFK (EL1,EL2,RHK,ZPKX,SGR,SGI)
C     HFK COMPUTES THE H FIELD OF A UNIFORM CURRENT FILAMENT BY
C     NUMERICAL INTEGRATION
      COMMON /TMH/ ZPK,RHKS
      DATA NX,NM,NTS,RX/1,65536,4,1.E-4/
      ZPK=ZPKX
      RHKS=RHK*RHK
      Z=EL1
      ZE=EL2
      S=ZE-Z
      EP=S/(10.*NM)
      ZEND=ZE-EP
      SGR=0.0
      SGI=0.0
      NS=NX
      NT=0
      CALL GH (Z,G1R,G1I)
1     DZ=S/NS
      ZP=Z+DZ
      IF (ZP-ZE) 3,3,2
2     DZ=ZE-Z
      IF (ABS(DZ)-EP) 17,17,3
3     DZOT=DZ*.5
      ZP=Z+DZOT
      CALL GH (ZP,G3R,G3I)
      ZP=Z+DZ
      CALL GH (ZP,G5R,G5I)
4     T00R=(G1R+G5R)*DZOT
      T00I=(G1I+G5I)*DZOT
      T01R=(T00R+DZ*G3R)*0.5
      T01I=(T00I+DZ*G3I)*0.5
      T10R=(4.0*T01R-T00R)/3.0
      T10I=(4.0*T01I-T00I)/3.0
      CALL TEST (T01R,T10R,TE1R,T01I,T10I,TE1I,0.)
      IF (TE1I-RX) 5,5,6
5     IF (TE1R-RX) 8,8,6
6     ZP=Z+DZ*0.25
      CALL GH (ZP,G2R,G2I)
      ZP=Z+DZ*0.75
      CALL GH (ZP,G4R,G4I)
      T02R=(T01R+DZOT*(G2R+G4R))*0.5
      T02I=(T01I+DZOT*(G2I+G4I))*0.5
      T11R=(4.0*T02R-T01R)/3.0
      T11I=(4.0*T02I-T01I)/3.0
      T20R=(16.0*T11R-T10R)/15.0
      T20I=(16.0*T11I-T10I)/15.0
      CALL TEST (T11R,T20R,TE2R,T11I,T20I,TE2I,0.)
      IF (TE2I-RX) 7,7,14
7     IF (TE2R-RX) 9,9,14
8     SGR=SGR+T10R
      SGI=SGI+T10I
      NT=NT+2
      GO TO 10
9     SGR=SGR+T20R
      SGI=SGI+T20I
      NT=NT+1
10    Z=Z+DZ
      IF (Z-ZEND) 11,17,17
11    G1R=G5R
      G1I=G5I
      IF (NT-NTS) 1,12,12
12    IF (NS-NX) 1,1,13
13    NS=NS/2
      NT=1
```

```
      GO TO 1
   14 NT=0
      IF (NS-NM) 16,15,15
   15 PRINT 18, Z
      GO TO 9
   16 NS=NS*2
      DZ=S/NS
      DZOT=DZ*0.5
      G5R=G3R
      G5I=G3I
      G3R=G2R
      G3I=G2I
      GO TO 4
   17 CONTINUE
      SGR=SGR*RHK*.5
      SGI=SGI*RHK*.5
      RETURN
C
   18 FORMAT (24H STEP SIZE LIMITED AT Z=,F10.5)
      END
```

HINTG

PURPOSE

To compute the near magnetic field due to a single patch in free space or over ground.

METHOD

The magnetic field is computed at the point, XI, YI, ZI due to the patch defined by parameters in COMMON/DATAJ/. The H field at $\bar{r} = (XI)\hat{x} + (YI)\hat{y} + (ZI)\hat{z}$ due to patch i, centered at $\bar{r}_i$, is approximated as:

$$\bar{H}(\bar{r}) = -\frac{1}{4\pi}\left[(1 + jkR)\frac{\exp(-jkR)}{(R/\lambda)^3}\right]\left[(\bar{R}/\lambda) \times \bar{J}_i\right] A_i/\lambda^2$$

where $\bar{R} = \bar{r} - \bar{r}_i$, and A_i is the area of patch i. This expression treats the surface currents as lumped at the center of the patch. $\bar{H}$ is computed for unit currents along the surface vectors $\hat{t}_{1i}$ and $\hat{t}_{2i}$.

When a ground is present, the code is executed twice in a loop. In the second pass, the field of the image of the patch is computed, multiplied by the reflection coefficients, and added to the direct field.

SYMBOL DICTIONARY

CR = cos (kR)

CTH = cos θ, θ = angle between the reflected ray and the normal to the ground

EXC, EYC, EZC = x, y, and z components of $\bar{H}$ excluding $(\times\ \bar{J}_i)$ term

EXK, EYK, EZK = $\bar{H}$ for $\bar{J}_i = \hat{t}_{1i}$

EXS, EYS, EZS = $\bar{H}$ for $\bar{J}_i = \hat{t}_{2i}$

F1X, F1Y, F1Z = $\bar{H}$ for $\bar{J}_i = \hat{t}_{1i}$; direct or reflected field contribution

F2X, F2Y, F2Z = $\overline{H}$ for $\overline{J}_i = \hat{t}_{2i}$; direct or reflected field contribution

FPI = 4π

GAM = $\overline{H}$ excluding the term $(\overline{R}/\lambda) \times \overline{J}_i$

IP = 1 for direct field, 2 for reflected field

IPERF = 1 for perfect ground, 0 otherwise

KSYMP = 1 for free space, 2 for ground

PX, PY = unit vector normal to plane of incidence for reflected ray $\hat{\rho}$

R = R/λ

RFL = +1 for direct field, -1 for reflected field

RK = kR; $k = 2\pi/\lambda$

RRH = R_H

RRV = R_V

RSQ = R^2/λ^2

RX, RY, RZ = $\overline{R}/\lambda$

S = A_i/λ^2

SR = $\sin(kR)$

T1XJ, T1YJ, T1ZJ = $\hat{t}_{1i}$

T2XJ, T2YJ, T2ZJ = $\hat{t}_{2i}$

T1ZR = z component of $\hat{t}_{1i}$ for patch i or for the image of patch i reflected in the ground

T2ZR = same as T1ZR for the $\hat{t}_{2i}$

TP = 2π

XI, YI, ZI = field evaluation point $\overline{r}/\lambda$

$$\left.\begin{matrix} XJ \\ YJ \\ ZJ \end{matrix}\right\} = \text{position of center of patch } \bar{r}_i/\lambda$$

XYMAG = magnitude of $\bar{R}/\lambda$ projected on the x-y plane

CONSTANTS

12.56637062 = 4π

6.283185308 = 2π

HINTG

```
      SUBROUTINE HINTG (XI,YI,ZI)
C     HINTG COMPUTES THE H FIELD OF A PATCH CURRENT
      COMPLEX EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC,ZRATI,ZRATI2,GAM,F1X,F
     11Y,F1Z,F2X,F2Y,F2Z,RRV,RRH,T1,FRATI
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y
     1J,IND1), (T2ZJ,IND2)
      DATA FPI/12.56637062/,TP/6.283185308/
      RX=XI-XJ
      RY=YI-YJ
      RFL=-1.
      EXK=(0.,0.)
      EYK=(0.,0.)
      EZK=(0.,0.)
      EXS=(0.,0.)
      EYS=(0.,0.)
      EZS=(0.,0.)
      DO 5 IP=1,KSYMP
      RFL=-RFL
      RZ=ZI-ZJ*RFL
      RSQ=RX*RX+RY*RY+RZ*RZ
      IF (RSQ.LT.1.E-20) GO TO 5
      R=SQRT(RSQ)
      RK=TP*R
      CR=COS(RK)
      SR=SIN(RK)
      GAM=-(CMPLX(CR,-SR)+RK*CMPLX(SR,CR))/(FPI*RSQ*R)*S
      EXC=GAM*RX
      EYC=GAM*RY
      EZC=GAM*RZ
      T1ZR=T1ZJ*RFL
      T2ZR=T2ZJ*RFL
      F1X=EYC*T1ZR-EZC*T1YJ
      F1Y=EZC*T1XJ-EXC*T1ZR
      F1Z=EXC*T1YJ-EYC*T1XJ
      F2X=EYC*T2ZR-EZC*T2YJ
      F2Y=EZC*T2XJ-EXC*T2ZR
      F2Z=EXC*T2YJ-EYC*T2XJ
      IF (IP.EQ.1) GO TO 4
      IF (IPERF.NE.1) GO TO 1
      F1X=-F1X
      F1Y=-F1Y
      F1Z=-F1Z
      F2X=-F2X
      F2Y=-F2Y
      F2Z=-F2Z
      GO TO 4
1     XYMAG=SQRT(RX*RX+RY*RY)
      IF (XYMAG.GT.1.E-6) GO TO 2
      PX=0.
      PY=0.
      CTH=1.
      RRV=(1.,0.)
      GO TO 3
2     PX=-RY/XYMAG
      PY=RX/XYMAG
      CTH=RZ/R
      RRV=CSQRT(1.-ZRATI*ZRATI*(1.-CTH*CTH))
3     RRH=ZRATI*CTH
      RRH=(RRH-RRV)/(RRH+RRV)
      RRV=ZRATI*RRV
```

```
      RRV=-(CTH-RRV)/(CTH+RRV)
      GAM=(F1X*PX+F1Y*PY)*(RRV-RRH)
      F1X=F1X*RRH+GAM*PX
      F1Y=F1Y*RRH+GAM*PY
      F1Z=F1Z*RRH
      GAM=(F2X*PX+F2Y*PY)*(RRV-RRH)
      F2X=F2X*RRH+GAM*PX
      F2Y=F2Y*RRH+GAM*PY
      F2Z=F2Z*RRH
    4 EXK=EXK+F1X
      EYK=EYK+F1Y
      EZK=EZK+F1Z
      EXS=EXS+F2X
      EYS=EYS+F2Y
      EZS=EZS+F2Z
    5 CONTINUE
      RETURN
      END
```

HSFLD

PURPOSE

To compute the near magnetic field due to constant, sine, and cosine current distributions on a segment in free space or over ground.

METHOD

The magnetic field is computed at the point XI, YI, ZI due to the segment defined by parameters in COMMON/DATAJ/. The fields computed by routine HSFLX are stored in /DATAJ/. When a ground is present, the code is executed twice in a loop. In the second pass, the field of the image of the segment is computed, multiplied by the reflection coefficients, and added to the direct field.

The field is evaluated in a cylindrical coordinate system with the source segment at the origin. The radius of a segment on which the field is evaluated is treated in the same way as for the electric field in subroutine EFLD. When the field evaluation point is not on a segment, the observation segment radius is set to zero in the call to HSFLD. Thus, as for the electric field, the ρ coordinate of the field evaluation point is computed for the surface of the observation segment as $\rho' = (\rho^2 + a^2)^{1/2}$, where ρ is the distance from the axis of the source segment to (XI, YI, ZI) and a is the radius of the observation segment. The resulting H field is multiplied by ρ/ρ'.

SYMBOL DICTIONARY

AI	= radius of observation segment, if any
CTH	= $\cos\theta$, θ = angle between the ray reflected from the ground and vertical
ETA	= $\eta = \sqrt{\mu/\epsilon}$
HPC, HPK, HPS	= H_ϕ due to cosine, constant, and sine current, respectively
PHX, PHY, PHZ	= $(\rho/\rho')\hat{\phi}$ in the cylindrical coordinates of the source segment or its image

PX, PY = unit vector normal to the plane of incidence of the reflected ray, $\hat{p}$

QX, QY, QZ = ρ/ρ' $[R_H\hat{\phi} + (R_V - R_H)(\hat{\phi} \cdot \hat{p})\hat{p}]$ for reflected ray

RFL = +1 for direct field, -1 for reflected field

RH = ρ'

RHOSPC = distance from coordinate origin to the point where the ray from the source to (XI, YI, ZI) reflects from the ground

RHOX, RHOY, RHOZ = $\overline{\rho}$ or $\overline{\rho}/\rho'$

RMAG = distance from the field evaluation point to the center of the source segment

RRH = R_H

RRV = R_V

SALPR = z component of unit vector in the direction of the source segment or its image

XI, YI, ZI = x, y, z coordinates of the field evaluation point

XIJ, YIJ, ZIJ = x, y, z components of distance from center of source segment to field observation point

XSPEC, YSPEC = x, y coordinates of the ground plane reflection point

XYMAG = horizontal distance from the source segment to the field observation point

ZP = projection of the vector (XIJ, YIJ, ZIJ) on the axis of the source segment

ZRATX = temporary storage for ZRATI

HSFLD

```
      SUBROUTINE HSFLD (XI,YI,ZI,AI)
C     HSFLD COMPUTES THE H FIELD FOR CONSTANT, SINE, AND COSINE CURRENT
C     ON A SEGMENT INCLUDING GROUND EFFECTS.
      COMPLEX EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC,ZRATI,ZRATI2,T1,HPK,HP
     1S,HPC,QX,QY,QZ,RRV,RRH,ZRATX,FRATI
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      DATA ETA/376.73/
      XIJ=XI-XJ
      YIJ=YI-YJ
      RFL=-1.
      DO 7 IP=1,KSYMP
      RFL=-RFL
      SALPR=SALPJ*RFL
      ZIJ=ZI-RFL*ZJ
      ZP=XIJ*CABJ+YIJ*SABJ+ZIJ*SALPR
      RHOX=XIJ-CABJ*ZP
      RHOY=YIJ-SABJ*ZP
      RHOZ=ZIJ-SALPR*ZP
      RH=SQRT(RHOX*RHOX+RHOY*RHOY+RHOZ*RHOZ+AI*AI)
      IF (RH.GT.1.E-10) GO TO 1
      EXK=0.
      EYK=0.
      EZK=0.
      EXS=0.
      EYS=0.
      EZS=0.
      EXC=0.
      EYC=0.
      EZC=0.
      GO TO 7
1     RHOX=RHOX/RH
      RHOY=RHOY/RH
      RHOZ=RHOZ/RH
      PHX=SABJ*RHOZ-SALPR*RHOY
      PHY=SALPR*RHOX-CABJ*RHOZ
      PHZ=CABJ*RHOY-SABJ*RHOX
      CALL HSFLX (S,RH,ZP,HPK,HPS,HPC)
      IF (IP.NE.2) GO TO 6
      IF (IPERF.EQ.1) GO TO 5
      ZRATX=ZRATI
      RMAG=SQRT(ZP*ZP+RH*RH)
      XYMAG=SQRT(XIJ*XIJ+YIJ*YIJ)
C
C     SET PARAMETERS FOR RADIAL WIRE GROUND SCREEN.
C
      IF (NRADL.EQ.0) GO TO 2
      XSPEC=(XI*ZJ+ZI*XJ)/(ZI+ZJ)
      YSPEC=(YI*ZJ+ZI*YJ)/(ZI+ZJ)
      RHOSPC=SQRT(XSPEC*XSPEC+YSPEC*YSPEC+T2*T2)
      IF (RHOSPC.GT.SCRWL) GO TO 2
      RRV=T1*RHOSPC*ALOG(RHOSPC/T2)
      ZRATX=(RRV*ZRATI)/(ETA*ZRATI+RRV)
2     IF (XYMAG.GT.1.E-6) GO TO 3
C
C     CALCULATION OF REFLECTION COEFFICIENTS WHEN GROUND IS SPECIFIED.
C
      PX=0.
      PY=0.
      CTH=1.
      RRV=(1.,0.)
      GO TO 4
```

```
    3  PX=-YIJ/XYMAG
       PY=XIJ/XYMAG
       CTH=ZIJ/RMAG
       RRV=CSQRT(1.-ZRATX*ZRATX*(1.-CTH*CTH))
    4  RRH=ZRATX*CTH
       RRH=-(RRH-RRV)/(RRH+RRV)
       RRV=ZRATX*RRV
       RRV=(CTH-RRV)/(CTH+RRV)
       QY=(PHX*PX+PHY*PY)*(RRV-RRH)
       QX=QY*PX+PHX*RRH
       QY=QY*PY+PHY*RRH
       QZ=PHZ*RRH
       EXK=EXK-HPK*QX
       EYK=EYK-HPK*QY
       EZK=EZK-HPK*QZ
       EXS=EXS-HPS*QX
       EYS=EYS-HPS*QY
       EZS=EZS-HPS*QZ
       EXC=EXC-HPC*QX
       EYC=EYC-HPC*QY
       EZC=EZC-HPC*QZ
       GO TO 7
    5  EXK=EXK-HPK*PHX
       EYK=EYK-HPK*PHY
       EZK=EZK-HPK*PHZ
       EXS=EXS-HPS*PHX
       EYS=EYS-HPS*PHY
       EZS=EZS-HPS*PHZ
       EXC=EXC-HPC*PHX
       EYC=EYC-HPC*PHY
       EZC=EZC-HPC*PHZ
       GO TO 7
    6  EXK=HPK*PHX
       EYK=HPK*PHY
       EZK=HPK*PHZ
       EXS=HPS*PHX
       EYS=HPS*PHY
       EZS=HPS*PHZ
       EXC=HPC*PHX
       EYC=HPC*PHY
       EZC=HPC*PHZ
    7  CONTINUE
       RETURN
       END
```

HSFLX

PURPOSE

To compute the near H field of filamentary currents of sine, cosine, and constant distribution on a segment.

METHOD

The wire segment is considered to be located at the origin of a local cylindrical coordinate system with the point at which the H field is computed being (ρ, ϕ, z). The coordinate geometry for a filament of current of length Δ is shown in figure 7. For a sine or cosine current distribution, the field can be written in closed form. For a current

$$I_0 \begin{bmatrix} \sin kz' \\ \cos kz' \end{bmatrix},$$

the field is

$$H_\phi(\rho, z) = \frac{-jI_0/\lambda}{2k\rho} \left\{ \exp(-jkr_2) \begin{bmatrix} \cos(k\Delta/2) \\ -\sin(k\Delta/2) \end{bmatrix} - \exp(-jkr_1) \begin{bmatrix} \cos(k\Delta/2) \\ \sin(k\Delta/2) \end{bmatrix} \right.$$

$$- j(kz - k\Delta/2) \frac{\exp(-jkr_2)}{kr_2} \begin{bmatrix} \sin(k\Delta/2) \\ \cos(k\Delta/2) \end{bmatrix}$$

$$\left. + j(kz + k\Delta/2) \frac{\exp(-jkr_1)}{kr_1} \begin{bmatrix} -\sin(k\Delta/2) \\ \cos(k\Delta/2) \end{bmatrix} \right\}.$$

$I_0/\lambda = 1$ is assumed in this routine.

For small values of ρ with $|z| > \Delta/2$, this equation may produce large numerical errors due to cancellation of large terms. Hence, for $z > 0$ and $\rho/(z - \Delta/2) < 10^{-3}$, a more stable approximation for small $\rho/(z \pm \Delta/2)$ is used:

$$H_\phi(\rho, z) = \frac{(\rho/\lambda)(Io/\lambda)}{8\pi} \exp(-jkz) \left\{ \left[\frac{2\pi}{(z + \Delta/2)/\lambda} - \frac{2\pi}{(z - \Delta/2)/\lambda} \right] \begin{bmatrix} 1 \\ -j \end{bmatrix} \right.$$

$$\left. + \left[\frac{\exp(jk\Delta/2)}{(z - \Delta/2)^2/\lambda^2} \begin{pmatrix} \sin(k\Delta/2) \\ \cos(k\Delta/2) \end{pmatrix} - \frac{\exp(-jk\Delta/2)}{(z + \Delta/2)^2/\lambda^2} \begin{pmatrix} -\sin(k\Delta/2) \\ \cos(k\Delta/2) \end{pmatrix} \right] \right\}.$$

For $z < 0$, the above equation is evaluated for $H_\phi(\rho, -z)$. The field of a $\sin kz'$ current is multiplied by -1 in this case, since it is an odd function of z.

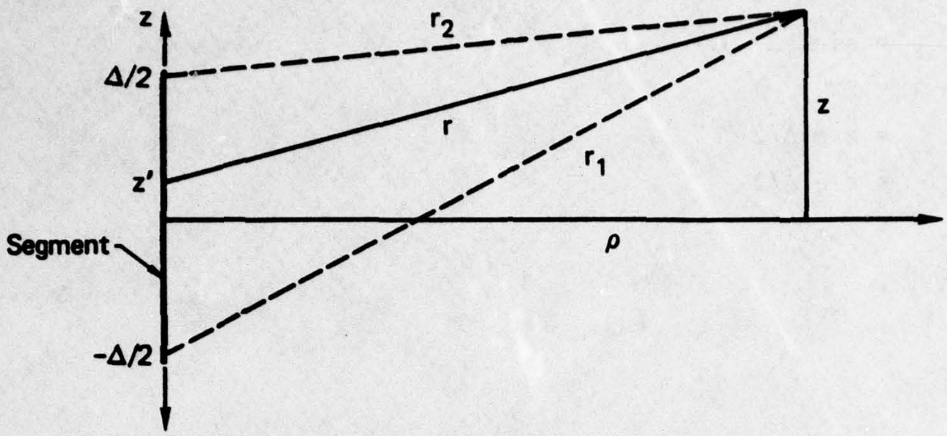


Figure 7. Coordinates for Evaluating H Field of a Segment.

The field due to a constant current is obtained by numerical integration, which is performed by subroutine HFK. If ρ is zero, all field quantities are set to zero, since H_ϕ is undefined.

SYMBOL DICTIONARY

CDK	= $\cos(k\Delta/2)$
CONS	= $-j/(2k\rho)$
DH	= $\Delta/2$
DK	= $k\Delta/2$
EKR1	= $\exp(-jkr_1)$
EKR2	= $\exp(-jkr_2)$
FJ	= j
FJK	= $-j2\pi$
HKR, HKI	= real and imaginary parts of H_ϕ due to a constant current
HPC, HPK, HPS	= H_ϕ due to cosine, constant, and sine currents, respectively
HSS	= sign of z
PI8	= 8π
R1	= r_1
R2	= r_2
RH	= ρ
RH2	= ρ^2
RHZ	= $\rho/(z - \Delta/2)$

S	=	Δ
SDK	=	$\sin(k\Delta/2)$
TP	=	2π
Z1	=	$z + \Delta/2$
Z2	=	$z - \Delta/2$
ZP	=	z

```
      SUBROUTINE HSFLX (S,RH,ZPX,HPK,HPS,HPC)
C     CALCULATES H FIELD OF SINE COSINE, AND CONSTANT CURRENT OF SEGMENT
      COMPLEX FJ,FJK,EKR1,EKR2,T1,T2,CONS,HPS,HPC,HPK
      DIMENSION FJX(2), FJKX(2)
      EQUIVALENCE (FJ,FJX), (FJK,FJKX)
      DATA TP/6.283185308/,FJX/0.,1./,FJKX/0.,-6.283185308/
      DATA PI8/25.13274123/
      IF (RH.LT.1.E-10) GO TO 6
      IF (ZPX.LT.0.) GO TO 1
      ZP=ZPX
      HSS=1.
      GO TO 2
    1 ZP=-ZPX
      HSS=-1.
    2 DH=.5*S
      Z1=ZP+DH
      Z2=ZP-DH
      IF (Z2.LT.1.E-7) GO TO 3
      RHZ=RH/Z2
      GO TO 4
    3 RHZ=1.
    4 DK=TP*DH
      CDK=COS(DK)
      SDK=SIN(DK)
      CALL HFK (-DK,DK,RH*TP,ZP*TP,HKR,HKI)
      HPK=CMPLX(HKR,HKI)
      IF (RHZ.LT.1.E-3) GO TO 5
      RH2=RH*RH
      R1=SQRT(RH2+Z1*Z1)
      R2=SQRT(RH2+Z2*Z2)
      EKR1=CEXP(FJK*R1)
      EKR2=CEXP(FJK*R2)
      T1=Z1*EKR1/R1
      T2=Z2*EKR2/R2
      HPS=(CDK*(EKR2-EKR1)-FJ*SDK*(T2+T1))*HSS
      HPC=-SDK*(EKR2+EKR1)-FJ*CDK*(T2-T1)
      CONS=-FJ/(2.*TP*RH)
      HPS=CONS*HPS
      HPC=CONS*HPC
      RETURN
    5 EKR1=CMPLX(CDK,SDK)/(Z2*Z2)
      EKR2=CMPLX(CDK,-SDK)/(Z1*Z1)
      T1=TP*(1./Z1-1./Z2)
      T2=CEXP(FJK*ZP)*RH/PI8
      HPS=T2*(T1+(EKR1+EKR2)*SDK)*HSS
      HPC=T2*(-FJ*T1+(EKR1-EKR2)*CDK)
      RETURN
    6 HPS=(0.,0.)
      HPC=(0.,0.)
      HPK=(0.,0.)
      RETURN
      END
```

INTRP

PURPOSE

To evaluate the Sommerfeld integral contributions to the field of a source over ground by interpolation in precomputed tables.

METHOD

The interpolation region in R_1 and θ is covered by three grids as shown in Figure 12 of Part I. The interpolation tables and the number of data points and the boundaries of each grid are read from file 21 and stored in COMMON/GGRID/ by the main program. In subroutine INTRP the variable x corresponds to R_1 and y to θ.

The three interpolation tables are stored in the arrays AR1, AR2 and AR3 in COMMON/GGRID/. For grid i, ARi(I,J,K) is the value at

$$x_I = s_i + (I - 1)\,\Delta x_i\,, \qquad I = 1, \ldots N_i$$

$$y_J = t_i + (J - 1)\,\Delta y_i\,, \qquad J = 1, \ldots M_i$$

where s_i = XSA(i) , Δx_i = DXA(i) , N_i = NXA(i)

t_i = YSA(i) , Δy_i = DYA(i) , M_i = NYA(i)

Each array contains values for I^V_ρ, I^V_z, I^H_ρ and I^H_ϕ from equations 156 through 159 of Part I for K equal to 1 through 4, respectively. The grid boundaries and density of points can be varied but the relative positions of the three grids must be as shown in Figure 12 of Part I for the logic for choosing the correct grid to work correctly. In particular, XSA(1), YSA(1) and YSA(2) must be zero; and XSA(2) and XSA(3) must be equal.

For a given x and y the values of I^V_ρ, I^V_z, I^H_ρ and I^H_ϕ are found by bivariate cubic interpolation and returned in the variables F1, F2, F3 and F4. The grid containing (x,y) is determined and a four by four point region containing (x,y) is selected. If x_i and y_k are the minimum values of x and y in the four by four point region then four interpolation polynomials in x are computed for $y = y_j$ with j = k, k + 1, k + 2, k + 3. These are

$$f_{ij}(x) = a_{ij}\xi_i^3 + b_{ij}\xi_i^2 + c_{ij}\xi_i + d_{ij}$$

where $\xi_i = (x - x_{i+1})/\Delta x$

$$a_{ij} = \frac{1}{6}\,[F_{i+3,j} - F_{i,j} + 3(F_{i+1,j} - F_{i+2,j})]$$

$$b_{ij} = \frac{1}{2}\,[F_{i,j} - 2F_{i+1,j} + F_{i+2,j}]$$

$$c_{ij} = F_{i+2,j} - \frac{1}{6}\,[2F_{i,j} + 3F_{i+1,j} + F_{i+3,j}]$$

$$d_{ij} = F_{i+1,j}$$

$$F_{i,j} = F(x_i, y_j)$$

A cubic polynomial in y, fit to the points $f_{ij}(x)$ for $j = k, \ldots k + 3$ is then evaluated for the given y to obtain the interpolated value $\hat{F}(x,y)$

$$\hat{F}(x,y) = \frac{1}{6}\,(p_1\eta_k^3 + p_2\eta_k^2 + p_3\eta_k) + p_4$$

$$\eta_k = (y - y_{k+1})/\Delta y$$

$$p_1 = f_{i,k+3}(x) - f_{ik}(x) + 3\,[f_{i,k+1}(x) - f_{i,k+2}(x)]$$

$$p_2 = 3[f_{i,k}(x) - 2f_{i,k+1}(x) + f_{i,k+2}(x)]$$

$$p_3 = 6f_{i,k+2}(x) - 2f_{i,k}(x) - 3f_{i,k+1}(x) - f_{i,k+3}(x)$$

$$p_4 = f_{i,k+1}$$

To reduce computation time the coefficients a_{ij}, b_{ij}, c_{ij} and d_{ij} are saved as long as successive points (x,y) fall in the same four by four point region of a grid. In addition the four by four point interpolation regions are restricted to starting indices i and k with values 3n + 1, n = 0, 1 .... Thus the regions do not overlap. This is less accurate than centering the region on each x,y point but requires less frequent computation of the coefficients. At the outer edges of a grid the regions are chosen to extend to the edge but not beyond. If x,y is out of the entire three grid region the nearest four by four point region is used for extrapolation.

The coefficients a_{ij}, b_{ij}, c_{ij} and d_{ij} are stored in two dimensional arrays from IT 106 to IT 109. When they are used, from IT 118 to

IT 149 they are used as simple variables ($A(1,1) \equiv A11$) to save time. Also the three dimensional arrays AR1, AR2, and AR3 are used as linear arrays from IT 92 to IT 105. The equivalent three subscripts are shown in the comment at IT 91.

SYMBOL DICTIONARY

Aij	= A(i,j) = a_{ij}
AR1	= ARL1 = grid 1
AR2	= ARL2 = grid 2
AR3	= ARL3 = grid 3
Bij	= B(i,j) = b_{ij}
Cij	= C(i,j) = c_{ij}
Dij	= D(i,j) = d_{ij}
DX	= Δx for grid being used
DXA	= array of Δx values for the three grids
DY	= Δy for grid being used
DYA	= array of Δy values
EPSCF	= $\epsilon_1 - j\sigma/\omega\epsilon_0$
F1	= I_ρ^V
F2	= I_Z^V
F3	= I_ρ^H
F4	= I_ϕ^H
FX1	= $f_{i,j}(x)$
FX2	= $f_{i,j+1}(x)$
FX3	= $f_{i,j+2}(x)$
FX4	= $f_{i,j+3}(x)$
IADD	= index for linear arrays ARL1, etc.
IADZ	= initial value for IADD
IGR	= grid number for present x,y
IGRS	= grid number for last x,y
IX	= x index of the grid coordinate just less than x
IXEG	= x index of the upper edge of the last normally located interpolation patch when a patch out of the

	normal locations is used at the outer edge of a grid, -10000 otherwise
IXS	= 1 plus the x index of the lower edge of 4 by 4 point interpolation patch
IY, IYEG, IYS	= same for y as IX, IXEG and IXS
K	= 1, 2, 3, 4 for I_ρ^V, I_z^V, I_ρ^H, I_ϕ^H
ND	= NDA for the particular grid
NDA	= array containing the first dimensions of AR1, AR2 and AR3
NDP	= NDPA for a particular grid
NDPA	= array containing the product of the first two dimensions in AR1, AR2 and AR3
NXA	= number of x values in each grid
NXM2	= NXA - 2 for a particular grid
NXMS	= upper x index of the last normally located patch at the edge of a grid
NYA, NYM2, NYMS	= same for y as NXA, NXM2 and NXMS
P1, P2, P3, P4	= p_1, p_2, p_3, p_4
X	= x
XS	= XSA for the present grid
XS2	= XSA(2) through equivalence
XSA	= array of values of x at lower edge of each grid (s_i)
XX	= ξ_i
XZ	= x_{i+1} for computing ξ_i
Y	= y
YS	= YSA for present grid
YS3	= YSA(3) through equivalence
YSA	= array of values of y at lower edge of each grid (t_i)
YY	= η_k
YZ	= y_{k+1} for computing η_k

INTRP

```
      SUBROUTINE INTRP (X,Y,F1,F2,F3,F4)
C
C     INTRP USES BIVARIATE CUBIC INTERPOLATION TO OBTAIN THE VALUES OF
C     4 FUNCTIONS AT THE POINT (X,Y).
C
      COMPLEX F1,F2,F3,F4,A,B,C,D,FX1,FX2,FX3,FX4,P1,P2,P3,P4,A11,A12,A1
     13,A14,A21,A22,A23,A24,A31,A32,A33,A34,A41,A42,A43,A44,B11,B12,B13,
     2B14,B21,B22,B23,B24,B31,B32,B33,B34,B41,B42,B43,B44,C11,C12,C13,C1
     34,C21,C22,C23,C24,C31,C32,C33,C34,C41,C42,C43,C44,D11,D12,D13,D14,
     4D21,D22,D23,D24,D31,D32,D33,D34,D41,D42,D43,D44
      COMPLEX AR1,AR2,AR3,ARL1,ARL2,ARL3,EPSCF
      COMMON /GGRID/ AR1(11,10,4),AR2(17,5,4),AR3(9,8,4),EPSCF,DXA(3),DY
     1A(3),XSA(3),YSA(3),NXA(3),NYA(3)
      DIMENSION NDA(3), NDPA(3)
      DIMENSION A(4,4), B(4,4), C(4,4), D(4,4), ARL1(1), ARL2(1), ARL3(1
     1)
      EQUIVALENCE (A(1,1),A11), (A(1,2),A12), (A(1,3),A13), (A(1,4),A14)
      EQUIVALENCE (A(2,1),A21), (A(2,2),A22), (A(2,3),A23), (A(2,4),A24)
      EQUIVALENCE (A(3,1),A31), (A(3,2),A32), (A(3,3),A33), (A(3,4),A34)
      EQUIVALENCE (A(4,1),A41), (A(4,2),A42), (A(4,3),A43), (A(4,4),A44)
      EQUIVALENCE (B(1,1),B11), (B(1,2),B12), (B(1,3),B13), (B(1,4),B14)
      EQUIVALENCE (B(2,1),B21), (B(2,2),B22), (B(2,3),B23), (B(2,4),B24)
      EQUIVALENCE (B(3,1),B31), (B(3,2),B32), (B(3,3),B33), (B(3,4),B34)
      EQUIVALENCE (B(4,1),B41), (B(4,2),B42), (B(4,3),B43), (B(4,4),B44)
      EQUIVALENCE (C(1,1),C11), (C(1,2),C12), (C(1,3),C13), (C(1,4),C14)
      EQUIVALENCE (C(2,1),C21), (C(2,2),C22), (C(2,3),C23), (C(2,4),C24)
      EQUIVALENCE (C(3,1),C31), (C(3,2),C32), (C(3,3),C33), (C(3,4),C34)
      EQUIVALENCE (C(4,1),C41), (C(4,2),C42), (C(4,3),C43), (C(4,4),C44)
      EQUIVALENCE (D(1,1),D11), (D(1,2),D12), (D(1,3),D13), (D(1,4),D14)
      EQUIVALENCE (D(2,1),D21), (D(2,2),D22), (D(2,3),D23), (D(2,4),D24)
      EQUIVALENCE (D(3,1),D31), (D(3,2),D32), (D(3,3),D33), (D(3,4),D34)
      EQUIVALENCE (D(4,1),D41), (D(4,2),D42), (D(4,3),D43), (D(4,4),D44)
      EQUIVALENCE (ARL1,AR1), (ARL2,AR2), (ARL3,AR3), (XS2,XSA(2)), (YS3
     1,YSA(3))
      DATA IXS,IYS,IGRS/-10,-10,-10/,DX,DY,XS,YS/1.,1.,0.,0./
      DATA NDA/11,17,9/,NDPA/110,85,72/,IXEG,IYEG/0,0/
      IF (X.LT.XS.OR.Y.LT.YS) GO TO 1
      IX=INT((X-XS)/DX)+1
      IY=INT((Y-YS)/DY)+1
C
C     IF POINT LIES IN SAME 4 BY 4 POINT REGION AS PREVIOUS POINT, OLD
C     VALUES ARE REUSED
C
      IF (IX.LT.IXEG.OR.IY.LT.IYEG) GO TO 1
      IF (IABS(IX-IXS).LT.2.AND.IABS(IY-IYS).LT.2) GO TO 12
C
C     DETERMINE CORRECT GRID AND GRID REGION
C
    1 IF (X.GT.XS2) GO TO 2
      IGR=1
      GO TO 3
    2 IGR=2
      IF (Y.GT.YS3) IGR=3
    3 IF (IGR.EQ.IGRS) GO TO 4
      IGRS=IGR
      DX=DXA(IGRS)
      DY=DYA(IGRS)
      XS=XSA(IGRS)
      YS=YSA(IGRS)
      NXM2=NXA(IGRS)-2
      NYM2=NYA(IGRS)-2
      NXMS=((NXM2+1)/3)*3+1
      NYMS=((NYM2+1)/3)*3+1
      ND=NDA(IGRS)
```

```
      NDP=NDPA(IGRS)
      IX=INT((X-XS)/DX)+1
      IY=INT((Y-YS)/DY)+1
    4 IXS=((IX-1)/3)*3+2
      IF (IXS.LT.2) IXS=2
      IXEG=-10000
      IF (IXS.LE.NXM2) GO TO 5
      IXS=NXM2
      IXEG=NXMS
    5 IYS=((IY-1)/3)*3+2
      IF (IYS.LT.2) IYS=2
      IYEG=-10000
      IF (IYS.LE.NYM2) GO TO 6
      IYS=NYM2
      IYEG=NYMS
C
C     COMPUTE COEFFICIENTS OF 4 CUBIC POLYNOMIALS IN X FOR THE 4 GRID
C     VALUES OF Y FOR EACH OF THE 4 FUNCTIONS
C
    6 IADZ=IXS+(IYS-3)*ND-NDP
      DO 11 K=1,4
      IADZ=IADZ+NDP
      IADD=IADZ
      DO 11 I=1,4
      IADD=IADD+ND
      GO TO (7,8,9), IGRS
C     P1=AR1(IXS-1,IYS-2+I,K)
    7 P1=ARL1(IADD-1)
      P2=ARL1(IADD)
      P3=ARL1(IADD+1)
      P4=ARL1(IADD+2)
      GO TO 10
    8 P1=ARL2(IADD-1)
      P2=ARL2(IADD)
      P3=ARL2(IADD+1)
      P4=ARL2(IADD+2)
      GO TO 10
    9 P1=ARL3(IADD-1)
      P2=ARL3(IADD)
      P3=ARL3(IADD+1)
      P4=ARL3(IADD+2)
   10 A(I,K)=(P4-P1+3.*(P2-P3))*.1666666667
      B(I,K)=(P1-2.*P2+P3)*.5
      C(I,K)=P3-(2.*P1+3.*P2+P4)*.1666666667
   11 D(I,K)=P2
      XZ=(IXS-1)*DX+XS
      YZ=(IYS-1)*DY+YS
C
C     EVALUATE POLYMOMIALS IN X AND THEN USE CUBIC INTERPOLATION IN Y
C     FOR EACH OF THE 4 FUNCTIONS.
C
   12 XX=(X-XZ)/DX
      YY=(Y-YZ)/DY
      FX1=((A11*XX+B11)*XX+C11)*XX+D11
      FX2=((A21*XX+B21)*XX+C21)*XX+D21
      FX3=((A31*XX+B31)*XX+C31)*XX+D31
      FX4=((A41*XX+B41)*XX+C41)*XX+D41
      P1=FX4-FX1+3.*(FX2-FX3)
      P2=3.*(FX1-2.*FX2+FX3)
      P3=6.*FX3-2.*FX1-3.*FX2-FX4
      F1=((P1*YY+P2)*YY+P3)*YY*.1666666667+FX2
      FX1=((A12*XX+B12)*XX+C12)*XX+D12
      FX2=((A22*XX+B22)*XX+C22)*XX+D22
      FX3=((A32*XX+B32)*XX+C32)*XX+D32
```

```
      FX4=((A42*XX+B42)*XX+C42)*XX+D42                                  IT 129
      P1=FX4-FX1+3.*(FX2-FX3)                                           IT 130
      P2=3.*(FX1-2.*FX2+FX3)                                            IT 131
      P3=6.*FX3-2.*FX1-3.*FX2-FX4                                       IT 132
      F2=((P1*YY+P2)*YY+P3)*YY*.1666666667+FX2                          IT 133
      FX1=((A13*XX+B13)*XX+C13)*XX+D13                                  IT 134
      FX2=((A23*XX+B23)*XX+C23)*XX+D23                                  IT 135
      FX3=((A33*XX+B33)*XX+C33)*XX+D33                                  IT 136
      FX4=((A43*XX+B43)*XX+C43)*XX+D43                                  IT 137
      P1=FX4-FX1+3.*(FX2-FX3)                                           IT 138
      P2=3.*(FX1-2.*FX2+FX3)                                            IT 139
      P3=6.*FX3-2.*FX1-3.*FX2-FX4                                       IT 140
      F3=((P1*YY+P2)*YY+P3)*YY*.1666666667+FX2                          IT 141
      FX1=((A14*XX+B14)*XX+C14)*XX+D14                                  IT 142
      FX2=((A24*XX+B24)*XX+C24)*XX+D24                                  IT 143
      FX3=((A34*XX+B34)*XX+C34)*XX+D34                                  IT 144
      FX4=((A44*XX+B44)*XX+C44)*XX+D44                                  IT 145
      P1=FX4-FX1+3.*(FX2-FX3)                                           IT 146
      P2=3.*(FX1-2.*FX2+FX3)                                            IT 147
      P3=6.*FX3-2.*FX1-3.*FX2-FX4                                       IT 148
      F4=((P1*YY+P2)*YY+P3)*YY*.1666666667+FX2                          IT 149
      RETURN                                                            IT 150
      END                                                               IT 151-
```

INTX

PURPOSE

To numerically compute the integral of the function exp(jkr)/kr.

METHOD

For evaluation of the field due to a segment, a local cylindrical coordinate system is defined with origin at the center of the segment and z axis in the segment direction. This geometry is illustrated in the discussion of subroutine GF. Subroutine INTX is called by subroutine EFLD to evaluate the integral

$$G = \int_{-k\Delta/2}^{k\Delta/2} \frac{\exp(-jkr)}{kr}\, d(kz) ,$$

where

$$r = [\rho'^2 + (z - z')^2]^{1/2} ,$$

and other symbols are defined in the discussion of subroutine GF.

The numerical integration technique of Romberg integration with variable interval width is used (refs. 3 and 4). The Romberg integration formula is obtained from the trapezoidal formula by an iterative procedure (ref. 1). The trapezoidal rule for integration of the function f(x) over an interval (a, b) using 2^k subintervals is

$$T_{0k} = [(b-a)/N][(1/2)\, f_0 + f_1 + \ldots + f_{N-1} + (1/2) f_N] ,$$

where

$$N = 2^k$$
$$f_i = f(x_i)$$
$$x_i = a + i(b - a)/N$$

These trapezoidal-rule answers are then used in the iterative formula

$$T_{m,n} = \left(4^m T_{m-1,n+1} - T_{m-1,n}\right)/(4^m - 1) .$$

The results $T_{m,n}$ may be arranged in a triangular matrix of the form

$$\begin{array}{lll} T_{0,0} & & \\ T_{0,1} & T_{1,0} & \\ T_{0,2} & T_{1,1} & T_{2,0} \\ \vdots & \vdots & \vdots \end{array}$$

where the elements in the first column, T_{0k}, represent the trapezoidal rule results, and the elements in the diagonal, T_{k0}, are the Romberg integration results for 2^k subintervals.

Convergence to increasingly more accurate answers takes place down the first column and the diagonal, as well as towards the right along the rows. The row convergence generally provides a more realistic indication of error magnitude than two successive trapezoidal-rule or Romberg answers.

This convergence along the rows is used to determine the interval width in the variable interval-width scheme. The complete integration interval is first divided into a minimum number of subintervals (presently set to 1) and T_{00}, T_{01}, and T_{10} are computed on the first subinterval. The relative difference of T_{01} and T_{10} is then computed, and if less than the error criterion, R_x, T_{10} is accepted as the integral over that interval, and integration proceeds to the next interval. If the difference of T_{01} and T_{10} is too great, T_{02}, T_{11} and T_{20} are computed. The relative difference of T_{11} and T_{20} is then computed, and if less than R_x, T_{20} is accepted as the integral over the subinterval. If the difference of T_{11} and T_{20} is too great, the subinterval is divided in half and the process repeated starting with T_{00} for the left hand, new subinterval. The subinterval is repeatedly halved until convergence to less than R_x is found. The process is repeated for successive subintervals until the right-hand side of the integration interval is reached. When convergence has been obtained with a given subinterval size for a few times, the routine attempts doubling the subinterval size to maintain the largest subinterval size that will give the required accuracy. Thus, the routine will use many points in a rapidly changing region of a function and fewer points where the function is smoothly varying.

Since the function to be integrated is complex, the convergence of both real and imaginary parts is tested and both must be less than R_x. The same subinterval sizes are used for real and imaginary parts.

When the field of a segment is being computed at the segment's own center, the length r becomes

$$r = [b^2 + (z - z')^2]^{1/2} ,$$

where b is the wire radius. For small values of b, the real part of the integrand is sharply peaked and, hence, difficult to integrate numerically. Hence, the integral is divided into the components

$$G' = \int_{-k\Delta/2}^{k\Delta/2} \frac{\exp(-jkr) - 1}{kr} \, d(kz)$$

$$G'' = \int_{-k\Delta/2}^{k\Delta/2} \frac{1}{kr} \, d(kz)$$

$$G = G' + G''$$

G' must be computed numerically; however, the integrand is no longer peaked. G", which contains the sharp peak, can be computed as

$$G'' = 2 \log \left(\frac{\sqrt{b^2 + \Delta^2} + \Delta}{b} \right)$$

To further reduce integration time for the self term, the integral of G' is computed from $-k\Delta/2$ to 0, and the result doubled to obtain G'.

SYMBOL DICTIONARY

ABS = external routine (absolute value)

ALOG = external routine (natural log)

B = wire radius, b/λ

DZ = subinterval size on which T_{00}, T_{01}, ... are computed

DZOT = 0.5 DZ

EL1 = $-k\Delta/2$

EL2 = $k\Delta/2$

EP = tolerance for ending the integration interval

FNM = real number equivalent of NM

FNS = real number equivalent of NS

GF = external routine (integrand)

G1I = imaginary part of f_1

G1R = real part of f_1

G2I = imaginary part of f_2
G2R = real part of f_2
G3I = imaginary part of f_3
G3R = real part of f_3
G4I = imaginary part of f_4
G4R = real part of f_4
G5I = imaginary part of f_5
G5R = real part of f_5
IJ = indication of self term integration when equal to zero
NM = minimum allowed subinterval size is $k\Delta$/NM
NS = present subinterval size is $k\Delta$/NS
NT = counter to control increasing of subinterval size
NTS = larger values retard increasing of subinterval size
NX = maximum allowed subinterval size is $k\Delta$/NX
RX = R_x
S = Δ/λ
SGI = imaginary part of G
SGR = real part of G
SQRT = external routine (square root)
TEST = external routine (computes relative convergence)
TE1I = relative difference of T_{01} and T_{10} for imaginary part
TE1R = relative difference of T_{01} and T_{10} for real part
TE2I = relative difference of T_{11} and T_{20} for imaginary part
TE2R = relative difference of T_{11} and T_{20} for real part
T00I = imaginary part T_{00}
T00R = real part T_{00}
T01I = imaginary part T_{01}
T01R = real part T_{01}
T02I = imaginary part T_{02}
T02R = real part T_{02}
T10I = imaginary part T_{10}
T10R = real part of T_{10}
T11I = imaginary part of T_{11}
T11R = real part of T_{11}
T20I = imaginary part of T_{20}
T20R = real part of T_{20}

Z = integration variable at left-hand side of subinterval

ZE = $k\Delta/2$

ZEND = $k\Delta/2$ - EP; EP = tolerance term

ZP = integration variable

CONSTANTS

65536 = 2^{16} = limit of minimum subinterval size (NM)

1.E-4 = error criterion, R_x

INTX

```
      SUBROUTINE INTX (EL1,EL2,B,IJ,SGR,SGI)
C
C     INTX PERFORMS NUMERICAL INTEGRATION OF EXP(JKR)/R BY THE METHOD OF
C     VARIABLE INTERVAL WIDTH ROMBERG INTEGRATION.  THE INTEGRAND VALUE
C     IS SUPPLIED BY SUBROUTINE GF.
C
      DATA NX,NM,NTS,RX/1,65536,4,1.E-4/
      Z=EL1
      ZE=EL2
      IF (IJ.EQ.0) ZE=0.
      S=ZE-Z
      FNM=NM
      EP=S/(10.*FNM)
      ZEND=ZE-EP
      SGR=0.
      SGI=0.
      NS=NX
      NT=0
      CALL GF (Z,G1R,G1I)
1     FNS=NS
      DZ=S/FNS
      ZP=Z+DZ
      IF (ZP-ZE) 3,3,2
2     DZ=ZE-Z
      IF (ABS(DZ)-EP) 17,17,3
3     DZOT=DZ*.5
      ZP=Z+DZOT
      CALL GF (ZP,G3R,G3I)
      ZP=Z+DZ
      CALL GF (ZP,G5R,G5I)
4     T00R=(G1R+G5R)*DZOT
      T00I=(G1I+G5I)*DZOT
      T01R=(T00R+DZ*G3R)*0.5
      T01I=(T00I+DZ*G3I)*0.5
      T10R=(4.0*T01R-T00R)/3.0
      T10I=(4.0*T01I-T00I)/3.0
C
C     TEST CONVERGENCE OF 3 POINT ROMBERG RESULT.
C
      CALL TEST (T01R,T10R,TE1R,T01I,T10I,TE1I,0.)
      IF (TE1I-RX) 5,5,6
5     IF (TE1R-RX) 8,8,6
6     ZP=Z+DZ*0.25
      CALL GF (ZP,G2R,G2I)
      ZP=Z+DZ*0.75
      CALL GF (ZP,G4R,G4I)
      T02R=(T01R+DZOT*(G2R+G4R))*0.5
      T02I=(T01I+DZOT*(G2I+G4I))*0.5
      T11R=(4.0*T02R-T01R)/3.0
      T11I=(4.0*T02I-T01I)/3.0
      T20R=(16.0*T11R-T10R)/15.0
      T20I=(16.0*T11I-T10I)/15.0
C
C     TEST CONVERGENCE OF 5 POINT ROMBERG RESULT.
C
      CALL TEST (T11R,T20R,TE2R,T11I,T20I,TE2I,0.)
      IF (TE2I-RX) 7,7,14
7     IF (TE2R-RX) 9,9,14
8     SGR=SGR+T10R
      SGI=SGI+T10I
      NT=NT+2
      GO TO 10
9     SGR=SGR+T20R
      SGI=SGI+T20I
```

```
      NT=NT+1
   10 Z=Z+DZ
      IF (Z-ZEND) 11,17,17
   11 G1R=G5R
      G1I=G5I
      IF (NT-NTS) 1,12,12
   12 IF (NS-NX) 1,1,13
C
C     DOUBLE STEP SIZE
C
   13 NS=NS/2
      NT=1
      GO TO 1
   14 NT=0
      IF (NS-NM) 16,15,15
   15 PRINT 20, Z
      GO TO 9
C
C     HALVE STEP SIZE
C
   16 NS=NS*2
      FNS=NS
      DZ=S/FNS
      DZOT=DZ*0.5
      G5R=G3R
      G5I=G3I
      G3R=G2R
      G3I=G2I
      GO TO 4
   17 CONTINUE
      IF (IJ) 19,18,19
C
C     ADD CONTRIBUTION OF NEAR SINGULARITY FOR DIAGONAL TERM
C
   18 SGR=2.*(SGR+ALOG((SQRT(B*B+S*S)+S)/B))
      SGI=2.*SGI
   19 CONTINUE
      RETURN
C
   20 FORMAT (24H STEP SIZE LIMITED AT Z=,F10.5)
      END
```

ISEGNO

PURPOSE

To determine the segment number of the m^{th} segment ordered by increasing segment numbers in the set of segments with tag numbers equal to the given tag number. With a given tag of zero, segment number m is returned.

METHOD

Search segments consecutively and check their tag numbers against a given tag.

SYMBOL DICTIONARY

I = DO loop index

ICNT = counter

ITAGI = input tag number (given tag)

M = input quantity specifying the position in the set of segments with the given tag

CODE LISTING

```
      FUNCTION ISEGNO (ITAGI,MX)
C
C     ISEGNO RETURNS THE SEGMENT NUMBER OF THE MTH SEGMENT HAVING THE
C     TAG NUMBER ITAGI.  IF ITAGI=0 SEGMENT NUMBER M IS RETURNED.
C
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      IF (MX.GT.0) GO TO 1
      PRINT 6
      STOP
1     ICNT=0
      IF (ITAGI.NE.0) GO TO 2
      ISEGNO=MX
      RETURN
2     IF (N.LT.1) GO TO 4
      DO 3 I=1,N
      IF (ITAG(I).NE.ITAGI) GO TO 3
      ICNT=ICNT+1
      IF (ICNT.EQ.MX) GO TO 5
3     CONTINUE
4     PRINT 7, ITAGI
      STOP
5     ISEGNO=I
      RETURN
C
6     FORMAT (4X,91HCHECK DATA, PARAMETER SPECIFYING SEGMENT POSITION IN
     1 A GROUP OF EQUAL TAGS MUST NOT BE ZERO)
7     FORMAT (///,10X,26HNO SEGMENT HAS AN ITAG OF ,I5)
      END
```

LFACTR

PURPOSE

To perform the Gauss-Doolittle factorization calculations on two blocks of the matrix in core storage. This routine in conjunction with FACIO factors a matrix that is too large for core storage into an upper and lower triangular matrix using the Gauss-Doolittle technique. The factored matrix is used by LUNSCR and LTSOLV to determine the solution of the transposed matrix equation $x^T A^T = B^T$.

METHOD

The basic algorithm used in this routine is presented by Ralston in ref. 1 on pages 411-416. A brief discussion is also given under FACTR in this manual. The main difference between LFACTR and FACTR is that LFACTR is set up to perform the calculations on two blocks of columns of the transposed matrix that reside in core storage. This situation arises when the matrix is too large to fit in core at one time; thus, the matrix is divided into blocks of columns and stored on files. This matrix is then factored into a lower triangular matrix and an upper triangular matrix by the subroutines FACIO and LFACTR. The function of these two subroutines is closely tied together: LFACTR performs the mathematical computations involved in the factorization, while FACIO controls the input and output of matrix blocks in core storage, and, thus, controls the necessary block ordering input to LFACTR. For clarification of the ordering of matrix blocks during factorization, refer to FACIO.

The computations performed in LFACTR are slightly different for three matrix block conditions: (1) block numbers 1 and 2, (2) adjacent matrix blocks, and (3) non-adjacent matrix blocks. If the blocks are numbers 1 and 2, both blocks are factored, and the computations proceed exactly as in FACTR. The only difference between LFACTR and FACTR here is that the two blocks do not form a square matrix, and the row and column indices in LFACTR have not been interchanged as in FACTR. At the end of this stage, both blocks 1 and 2 are completely factored. For case 2, where the blocks are adjacent in the matrix and other than 1 and 2, the first block is assumed factored and is used to complete the factorization of the partially factored second block. The computations start with the first column of the second block and proceed as in FACTR (with the exceptions noted above). If the blocks are not adjacent (case 3), the first block is assumed factored and is used to partially

factor the second block. Computations start with the first column of the second block. Factorization cannot be completed, since values from the intervening columns are necessary.

CODING

LF20 - LF39 Initialization of loop parameters for the various matrix block conditions.

LF40 - LF99 Loop over columns to be factored or partially factored.

LF44 - LF46 Write column of A in scratch vector D.

LF49 - LF62 Computations for u_{ir} (see FACTR), where positioning for size is taken into account. The range of i is determined by the matrix blocks used.

LF69 - LF71 For case 3, the partially factored column is stored in A, and a jump to LF100 is made.

LF73 - LF87 For cases 1 and 2, the maximum value in the column is found for positioning.

LF92 - LF94 For cases 1 and 2, ℓ_{ir} (see FACTR) is calculated; limits on i are dependent on blocks.

SYMBOL DICTIONARY

A = array which contains the two blocks of columns of the transposed matrix in some state of factorization

CONJG = external routine (conjugate of complex numbers)

D = scratch vector, temporary storage of one column

DMAX = maximum value in column

ELMAG = intermediate variable

I = DO loop index

IFLG = small pivot value flag

IP = array containing positioning information

IXJ = index

IX1 = first block number, input

IX2 = second block number, input

J = DO loop index

JP1 = J + 1

J1, J2 = DO loop limits

J2P1 = J2 + 1

J2P2 = J2 + 2

K = DO loop index

L1, L2, L3 = logical variables for testing

NCOL = number of columns

NROW = number of rows

PJ, PR = intermediate variables

R = DO loop index

REAL = external routine (real part of a complex number)

R1, R2 = DO loop limits, relative column number limits for calculations

In programs using double precision accumulation in the matrix solution, the following double precision variables are used in LFACTR.

DAR1, DAI1, DAR2, DAI2 = real and imaginary parts of a number for temporary storage

DR, DI = real and imaginary vectors replacing the complex vector D in single precision programs

CONSTANT

1.E-10 = small value test

```
      SUBROUTINE LFACTR (A,NROW,IX1,IX2,IP)                             LF   1
C                                                                       LF   2
C     LFACTR PERFORMS GAUSS-DOOLITTLE MANIPULATIONS ON THE TWO BLOCKS OF LF   3
C     THE TRANSPOSED MATRIX IN CORE STORAGE.  THE GAUSS-DOOLITTLE       LF   4
C     ALGORITHM IS PRESENTED ON PAGES 411-416 OF A. RALSTON -- A FIRST  LF   5
C     COURSE IN NUMERICAL ANALYSIS.  COMMENTS BELOW REFER TO COMMENTS IN LF   6
C     RALSTONS TEXT.                                                    LF   7
C                                                                       LF   8
      COMPLEX A,D,AJR                                                   LF   9
      INTEGER R,R1,R2,PJ,PR                                             LF  10
      LOGICAL L1,L2,L3                                                  LF  11
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I LF  12
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL                                LF  13
      COMMON /SCRATM/ D(600)                                            LF  14
      DIMENSION A(NROW,1), IP(NROW)                                     LF  15
      IFLG=0                                                            LF  16
C                                                                       LF  17
C     INITIALIZE R1,R2,J1,J2                                            LF  18
C                                                                       LF  19
      L1=IX1.EQ.1.AND.IX2.EQ.2                                          LF  20
      L2=(IX2-1).EQ.IX1                                                 LF  21
      L3=IX2.EQ.NBLSYM                                                  LF  22
      IF (L1) GO TO 1                                                   LF  23
      GO TO 2                                                           LF  24
1     R1=1                                                              LF  25
      R2=2*NPSYM                                                        LF  26
      J1=1                                                              LF  27
      J2=-1                                                             LF  28
      GO TO 5                                                           LF  29
2     R1=NPSYM+1                                                        LF  30
      R2=2*NPSYM                                                        LF  31
      J1=(IX1-1)*NPSYM+1                                                LF  32
      IF (L2) GO TO 3                                                   LF  33
      GO TO 4                                                           LF  34
3     J2=J1+NPSYM-2                                                     LF  35
      GO TO 5                                                           LF  36
4     J2=J1+NPSYM-1                                                     LF  37
5     IF (L3) R2=NPSYM+NLSYM                                            LF  38
      DO 16 R=R1,R2                                                     LF  39
C                                                                       LF  40
C     STEP 1                                                            LF  41
C                                                                       LF  42
      DO 6 K=J1,NROW                                                    LF  43
      D(K)=A(K,R)                                                       LF  44
6     CONTINUE                                                          LF  45
C                                                                       LF  46
C     STEPS 2 AND 3                                                     LF  47
C                                                                       LF  48
      IF (L1.OR.L2) J2=J2+1                                             LF  49
      IF (J1.GT.J2) GO TO 9                                             LF  50
      IXJ=0                                                             LF  51
      DO 8 J=J1,J2                                                      LF  52
      IXJ=IXJ+1                                                         LF  53
      PJ=IP(J)                                                          LF  54
      AJR=D(PJ)                                                         LF  55
      A(J,R)=AJR                                                        LF  56
      D(PJ)=D(J)                                                        LF  57
      JP1=J+1                                                           LF  58
      DO 7 I=JP1,NROW                                                   LF  59
      D(I)=D(I)-A(I,IXJ)*AJR                                            LF  60
7     CONTINUE                                                          LF  61
8     CONTINUE                                                          LF  62
9     CONTINUE                                                          LF  63
C                                                                       LF  64
```

```
C       STEP 4
C
        J2P1=J2+1
        IF (L1.OR.L2) GO TO 11
        IF (NROW.LT.J2P1) GO TO 16
        DO 10 I=J2P1,NROW
        A(I,R)=D(I)
   10   CONTINUE
        GO TO 16
   11   DMAX=REAL(D(J2P1)*CONJG(D(J2P1)))
        IP(J2P1)=J2P1
        J2P2=J2+2
        IF (J2P2.GT.NROW) GO TO 13
        DO 12 I=J2P2,NROW
        ELMAG=REAL(D(I)*CONJG(D(I)))
        IF (ELMAG.LT.DMAX) GO TO 12
        DMAX=ELMAG
        IP(J2P1)=I
   12   CONTINUE
   13   CONTINUE
        IF (DMAX.LT.1.E-10) IFLG=1
        PR=IP(J2P1)
        A(J2P1,R)=D(PR)
        D(PR)=D(J2P1)
C
C       STEP 5
C
        IF (J2P2.GT.NROW) GO TO 15
        AJR=1./A(J2P1,R)
        DO 14 I=J2P2,NROW
        A(I,R)=D(I)*AJR
   14   CONTINUE
   15   CONTINUE
        IF (IFLG.EQ.0) GO TO 16
        PRINT 17, J2,DMAX
        IFLG=0
   16   CONTINUE
        RETURN
C
   17   FORMAT (1H ,6HPIVOT(,I3,2H)=,E16.8)
        END
```

LOAD

PURPOSE

To compute the impedances at a given frequency for the loading specified by LD cards.

METHOD

The value of $\lambda Z/\Delta$, where Z is the total impedance on a segment and Δ is the length of the segment, is computed for each loaded segment and stored in the array ZARRAY. The proper impedance formula is chosen by the value of the input quantity LDTYP. These computations are performed from the sequence L074 to L096 of the program, and the formulas are:

LDTYP = 0 (series R, L, and C):

$$Z = R + j\omega L + \frac{1}{j\omega C}$$

$$Z' = \frac{\lambda Z}{\Delta} = \frac{R}{\frac{\Delta}{\lambda}} + j2\pi c\left(\frac{L}{\Delta}\right) + \frac{1}{j2\pi c\left(\frac{\Delta}{\lambda}\right)^2\left(\frac{C}{\Delta}\right)}$$

where c is the speed of light and R, L, and C are input.

LDTYP = 1 (parallel R, L, and C; R, L, and C input):

$$Z' = \frac{1}{\left(\frac{\Delta}{\lambda}\right)\frac{1}{R} + \frac{\Delta}{j2\pi cL} + j2\pi c\left(\frac{\Delta}{\lambda}\right)^2\left(\frac{C}{\Delta}\right)}$$

LDTYP = 2 and 3 (same as above, but R/Δ, L/Δ, C/Δ are input)

LDTYP = 4 (resistance and reactance input):

$$Z' = \frac{\text{resistance} + j\ \text{reactance}}{\frac{\Delta}{\lambda}}$$

LDTYP = 5 (call another subroutine for wire conductivity calculation)

SYMBOL DICTIONARY

ABS = external routine (absolute value of a real number)

AIMAG = external routine (imaginary part of a complex number)

CMPLX = external routine (forms a complex number)

ICHK = check flag in diagnosing data errors

ISTEP = loading card subscript

IWARN = flag checking for multiply loaded segments

JUMP = LDTYP + 1

LDTAG = tag number, input quantity

LDTAGF = input quantity

LDTAGS = LDTAG(ISTEP)

LDTAGT = input quantity

LDTYP = input quantity specifying loading type

NLOAD = number of input loading data cards

PRNT = external routine (prints the impedance data in a table)

REAL = external routine (takes the real part of a complex number)

TPCJ = $j2\pi c$, where c is the speed of light

ZARRAY = array containing $\lambda Z/\Delta$ for each segment, dimensioned to the maximum number of segments

ZINT = external routine (calculates the internal impedance of a finitely conducting wire)

ZLC, ZLI, ZLR = input quantities, the definitions are a function of the type of loading specified. For the case of series RLC (LDTYP = 0): ZLC = capacitance (farads), ZLI = inductance (henrys), and ZLR = resistance (ohms). For the remaining cases, see Part III.

ZT = $Z' = \lambda Z/\Delta$ for one segment; however, variable name is used during the calculation of this quantity

CONSTANTS

1.E-20 = floating point zero test

(0., 1.88365371E+9) = $j2\pi c$, where c is the velocity of light

```
      SUBROUTINE LOAD (LDTYP,LDTAG,LDTAGF,LDTAGT,ZLR,ZLI,ZLC)
C
C     LOAD CALCULATES THE IMPEDANCE OF SPECIFIED SEGMENTS FOR VARIOUS
C     TYPES OF LOADING
C
      COMPLEX ZARRAY,ZT,TPCJ,ZINT
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF
      DIMENSION LDTYP(1), LDTAG(1), LDTAGF(1), LDTAGT(1), ZLR(1), ZLI(1)
     1, ZLC(1), TPCJX(2)
      EQUIVALENCE (TPCJ,TPCJX)
      DATA TPCJX/0.,1.883698955E+9/
C
C     PRINT HEADING
C
      PRINT 25
C
C     INITIALIZE D ARRAY, USED FOR TEMPORARY STORAGE OF LOADING
C     INFORMATION.
C
      DO 1 I=N2,N
1     ZARRAY(I)=(0.,0.)
      IWARN=0
C
C     CYCLE OVER LOADING CARDS
C
      ISTEP=0
2     ISTEP=ISTEP+1
      IF (ISTEP.LE.NLOAD) GO TO 5
      IF (IWARN.EQ.1) PRINT 26
      IF (N1+2*M1.GT.0) GO TO 4
      NOP=N/NP
      IF (NOP.EQ.1) GO TO 4
      DO 3 I=1,NP
      ZT=ZARRAY(I)
      L1=I
      DO 3 L2=2,NOP
      L1=L1+NP
3     ZARRAY(L1)=ZT
4     RETURN
5     IF (LDTYP(ISTEP).LE.5) GO TO 6
      PRINT 27, LDTYP(ISTEP)
      STOP
6     LDTAGS=LDTAG(ISTEP)
      JUMP=LDTYP(ISTEP)+1
      ICHK=0
C
C     SEARCH SEGMENTS FOR PROPER ITAGS
C
      L1=N2
      L2=N
      IF (LDTAGS.NE.0) GO TO 7
      IF (LDTAGF(ISTEP).EQ.0.AND.LDTAGT(ISTEP).EQ.0) GO TO 7
      L1=LDTAGF(ISTEP)
      L2=LDTAGT(ISTEP)
      IF (L1.GT.N1) GO TO 7
      PRINT 29
      STOP
7     DO 17 I=L1,L2
      IF (LDTAGS.EQ.0) GO TO 8
      IF (LDTAGS.NE.ITAG(I)) GO TO 17
```

```
      IF (LDTAGF(ISTEP).EQ.0) GO TO 8                                    LO  64
      ICHK=ICHK+1                                                        LO  65
      IF (ICHK.GE.LDTAGF(ISTEP).AND.ICHK.LE.LDTAGT(ISTEP)) GO TO 9       LO  66
      GO TO 17                                                           LO  67
    8 ICHK=1                                                             LO  68
C                                                                        LO  69
C     CALCULATION OF LAMDA*IMPED. PER UNIT LENGTH, JUMP TO APPROPRIATE   LO  70
C     SECTION FOR LOADING TYPE                                           LO  71
C                                                                        LO  72
    9 GO TO (10,11,12,13,14,15), JUMP                                    LO  73
   10 ZT=ZLR(ISTEP)/SI(I)+TPCJ*ZLI(ISTEP)/(SI(I)*WLAM)                   LO  74
      IF (ABS(ZLC(ISTEP)).GT.1.E-20) ZT=ZT+WLAM/(TPCJ*SI(I)*ZLC(ISTEP))  LO  75
      GO TO 16                                                           LO  76
   11 ZT=TPCJ*SI(I)*ZLC(ISTEP)/WLAM                                      LO  77
      IF (ABS(ZLI(ISTEP)).GT.1.E-20) ZT=ZT+SI(I)*WLAM/(TPCJ*ZLI(ISTEP))  LO  78
      IF (ABS(ZLR(ISTEP)).GT.1.E-20) ZT=ZT+SI(I)/ZLR(ISTEP)              LO  79
      ZT=1./ZT                                                           LO  80
      GO TO 16                                                           LO  81
   12 ZT=ZLR(ISTEP)*WLAM+TPCJ*ZLI(ISTEP)                                 LO  82
      IF (ABS(ZLC(ISTEP)).GT.1.E-20) ZT=ZT+1./(TPCJ*SI(I)*SI(I)*ZLC(ISTE LO  83
     1P))                                                                LO  84
      GO TO 16                                                           LO  85
   13 ZT=TPCJ*SI(I)*SI(I)*ZLC(ISTEP)                                     LO  86
      IF (ABS(ZLI(ISTEP)).GT.1.E-20) ZT=ZT+1./(TPCJ*ZLI(ISTEP))          LO  87
      IF (ABS(ZLR(ISTEP)).GT.1.E-20) ZT=ZT+1./(ZLR(ISTEP)*WLAM)          LO  88
      ZT=1./ZT                                                           LO  89
      GO TO 16                                                           LO  90
   14 ZT=CMPLX(ZLR(ISTEP),ZLI(ISTEP))/SI(I)                              LO  91
      GO TO 16                                                           LO  92
   15 ZT=ZINT(ZLR(ISTEP)*WLAM,BI(I))                                     LO  93
   16 IF ((ABS(REAL(ZARRAY(I)))+ABS(AIMAG(ZARRAY(I)))).GT.1.E-20) IWARN= LO  94
     11                                                                  LO  95
      ZARRAY(I)=ZARRAY(I)+ZT                                             LO  96
   17 CONTINUE                                                           LO  97
      IF (ICHK.NE.0) GO TO 18                                            LO  98
      PRINT 28, LDTAGS                                                   LO  99
      STOP                                                               LO 100
C                                                                        LO 101
C     PRINTING THE SEGMENT LOADING DATA, JUMP TO PROPER PRINT            LO 102
C                                                                        LO 103
   18 GO TO (19,20,21,22,23,24), JUMP                                    LO 104
   19 CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP LO 105
     1),ZLC(ISTEP),0.,0.,0.,7H SERIES,7)                                 LO 106
      GO TO 2                                                            LO 107
   20 CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP LO 108
     1),ZLC(ISTEP),0.,0.,0.,8HPARALLEL,8)                                LO 109
      GO TO 2                                                            LO 110
   21 CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP LO 111
     1),ZLC(ISTEP),0.,0.,0.,18HSERIES (PER METER),18)                    LO 112
      GO TO 2                                                            LO 113
   22 CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),ZLR(ISTEP),ZLI(ISTEP LO 114
     1),ZLC(ISTEP),0.,0.,0.,20HPARALLEL (PER METER),20)                  LO 115
      GO TO 2                                                            LO 116
   23 CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),0.,0.,0.,ZLR(ISTEP), LO 117
     1ZLI(ISTEP),0.,15HFIXED IMPEDANCE,15)                               LO 118
      GO TO 2                                                            LO 119
   24 CALL PRNT (LDTAGS,LDTAGF(ISTEP),LDTAGT(ISTEP),0.,0.,0.,0.,0.,ZLR(I LO 120
     1STEP),6H  WIRE,6)                                                  LO 121
      GO TO 2                                                            LO 122
C                                                                        LO 123
   25 FORMAT (//,7X,8HLOCATION,10X,10HRESISTANCE,3X,10HINDUCTANCE,2X,11H LO 124
     1CAPACITANCE,7X,16HIMPEDANCE (OHMS),5X,12HCONDUCTIVITY,4X,4HTYPE,/, LO 125
     24X,4HITAG,10H FROM THRU,10X,4HOHMS,8X,6HHENRYS,7X,6HFARADS,8X,4HRE LO 126
     3AL,6X,9HIMAGINARY,4X,10HMHOS/METER)                                LO 127
```

```
   26  FORMAT (/,10X,74HNOTE, SOME OF THE ABOVE SEGMENTS HAVE BEEN LOADED LO 128
      1 TWICE - IMPEDANCES ADDED)                                         LO 129
   27  FORMAT (/,10X,46HIMPROPER LOAD TYPE CHOOSEN, REQUESTED TYPE IS ,I3 LO 130
      1)                                                                  LO 131
   28  FORMAT (/,10X,50HLOADING DATA CARD ERROR, NO SEGMENT HAS AN ITAG = LO 132
      1 ,I5)                                                              LO 133
   29  FORMAT (63H ERROR - LOADING MAY NOT BE ADDED TO SEGMENTS IN N.G.F. LO 134
      1 SECTION)                                                          LO 135
       END                                                                LO 136-
```

LTSOLV

PURPOSE

To solve the matrix equation $X^R LU = B^R$, where R denotes a row vector and L and U are the lower and upper triangular matrices stored as blocks on files.

METHOD

The L and U triangular matrices are written in a square array, where the 1's on the diagonal of the L matrix are suppressed. The array is stored by blocks of columns in ascending order on file IFL1 and descending order on file IFL2. The solution procedure is as follows. First solve the equation

$$Y^R U = B^R \tag{1}$$

then

$$X^R L = Y^R , \tag{2}$$

since $X^R LU = B^R$. The solutions of equations (1) and (2) are straightforward, since both matrices are triangular. In particular for equation (1),

$$y_j^R = \frac{1}{u_{jj}} \left(b_j^R - \sum_{i=1}^{j-1} y_i^R u_{ij} \right) \qquad j = 1, \ldots, n$$

and similarly for equation (2).

Several right-hand side vectors may be stored in the two dimensional array B. The forward and backward substitution is then done on each vector in the loops from LT 23 to LT 34 and LT 43 to LT 56. This can be much faster than calling LTSOLV for each vector since the files IFL1 and IFL2 are read only once. This feature is used in computing $A^{-1}B$ for the NGF solution. It is not used with the multiple excitations for a receiving pattern or to compute the driving point interaction matrix in NETWK but could reduce the out-of-core solution time in these cases.

Row interchanges were used to position elements for size in factoring the transposed structure matrix; therefore, the elements in the solution vector X^R are not in the original locations. Using the IX array (filled by LUNSCR), the vector can be put back into the original order. The integer contained in IX(J) is the index of the original location of the parameter now in the j^{th} location. The solution vector is overwritten on the input right-hand side vector B^R.

SYMBOL DICTIONARY

A	= array for matrix blocks
B	= B^R, right-hand side and solution
I2	= number of words in a block
IFL1	= file with blocks in normal order
IFL2	= file with blocks in reversed order
IX	= solution unscramble vector
IXBLK1	= block number
J	= row index
JST	= initial value for J
K2	= number of columns in a block
KP	= column index
NEQ	= total number of equations
NRH	= number of right-hand side vectors in B
NROW	= row dimension of A (number of equations in a symmetric section)
SUM	= summation result

```
      SUBROUTINE LTSOLV (A,NROW,IX,B,NEQ,NRH,IFL1,IFL2)
C
C     LTSOLV SOLVES THE MATRIX EQ. Y(R)*LU(T)=B(R) WHERE (R) DENOTES ROW
C     VECTOR AND LU(T) DENOTES THE LU DECOMPOSITION OF THE TRANSPOSE OF
C     THE ORIGINAL COEFFICIENT MATRIX.  THE LU(T) DECOMPOSITION IS
C     STORED ON TAPE 5 IN BLOCKS IN ASCENDING ORDER AND ON FILE 3 IN
C     BLOCKS OF DESCENDING ORDER.
C
      COMPLEX A,B,Y,SUM
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      COMMON /SCRATM/ Y(600)
      DIMENSION A(NROW,NROW), B(NEQ,NRH), IX(NEQ)
C
C     FORWARD SUBSTITUTION
C
      I2=2*NPSYM*NROW
      DO 4 IXBLK1=1,NBLSYM
      CALL BLCKIN (A,IFL1,1,I2,1,121)
      K2=NPSYM
      IF (IXBLK1.EQ.NBLSYM) K2=NLSYM
      JST=(IXBLK1-1)*NPSYM
      DO 4 IC=1,NRH
      J=JST
      DO 3 K=1,K2
      JM1=J
      J=J+1
      SUM=(0.,0.)
      IF (JM1.LT.1) GO TO 2
      DO 1 I=1,JM1
1     SUM=SUM+A(I,K)*B(I,IC)
2     B(J,IC)=(B(J,IC)-SUM)/A(J,K)
3     CONTINUE
4     CONTINUE
C
C     BACKWARD SUBSTITUTION
C
      JST=NROW+1
      DO 8 IXBLK1=1,NBLSYM
      CALL BLCKIN (A,IFL2,1,I2,1,122)
      K2=NPSYM
      IF (IXBLK1.EQ.1) K2=NLSYM
      DO 7 IC=1,NRH
      KP=K2+1
      J=JST
      DO 6 K=1,K2
      KP=KP-1
      JP1=J
      J=J-1
      SUM=(0.,0.)
      IF (NROW.LT.JP1) GO TO 6
      DO 5 I=JP1,NROW
5     SUM=SUM+A(I,KP)*B(I,IC)
      B(J,IC)=B(J,IC)-SUM
6     CONTINUE
7     CONTINUE
8     JST=JST-K2
C
C     UNSCRAMBLE SOLUTION
C
      DO 10 IC=1,NRH
      DO 9 I=1,NROW
      IXI=IX(I)
9     Y(IXI)=B(I,IC)
```

```
      DO 10 I=1,NROW
10    B(I,IC)=Y(I)
      RETURN
      END
```

LUNSCR

PURPOSE

To unscramble the lower triangular matrix of the factored out-of-core matrix and to determine the appropriate ordering of the unknowns. The unscrambled factored matrix is written in blocks on file IU3 in ascending order and on file IU4 in descending order.

METHOD

During factorization by LFACTR, the elements in the lower triangular matrix L were not explicitly arranged in accordance with the row interchanges used in positioning for size during the calculations. Specifically, as the factorization proceeds by columns from left to right in the matrix, row rearrangements in the r^{th} column are not explicitly performed in the left $r - 1$ columns; rather, positioning information is stored in the IP array. For the in-core calculations, these rearrangements are included during the final solution (subroutine SOLVE). For the out-of-core case, rearrangement during the solution (subroutine LTSOLV) is inconvenient, since the transposed system $x^r A^t = B^r$ is being solved, where r signifies a row vector.

The procedure for unscrambling the L matrix is as follows. p_k is the positioning information contained in IP(K). Then for the r^{th} column, let t be a temporary variable:

$$t = \ell_{k,r}$$

$$\ell_{p_k,r} \text{ overwrites } \ell_{k,r}$$

$$t \text{ overwrites } \ell_{p_k,r} \qquad \text{for } k = r + 1, \ldots, n - 1$$

Since row interchanges were used on the transposed matrix, the positions of the unknowns in the equations have changed. The final arrangement is determined by performing interchanges on a vector of integers. Specifically, let

$$x_i = i \qquad i = 1, \ldots, n$$

then set

$$t = x_k$$

x_{p_k} overwrites x_k

t overwrites x_{p_k} for $k = 1, \ldots, n$

The integer now contained in x_i specifies the original placement of the i^{th} unknown.

SYMBOL DICTIONARY

A	= array for matrix blocks
I1	= first word of matrix block
I2	= last word of matrix block
IP	= array of pivot index data
IU2	= input file
IU3	= output file, blocks in normal order
IU4	= output file, blocks in reversed order
IX	= array x_i
IXBLK1	= block number
KA	= *increment to locate the* KK^{th} submatrix in case of symmetry
NOP	= number of symmetric sections
NROW	= row dimension of A

```
      SUBROUTINE LUNSCR (A,NROW,NOP,IX,IP,IU2,IU3,IU4)
C
C     S/R WHICH UNSCRAMBLES, SCRAMBLED FACTORED MATRIX
C
      COMPLEX A,TEMP
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      DIMENSION A(NROW,1), IP(NROW), IX(NROW)
      I1=1
      I2=2*NPSYM*NROW
      NM1=NROW-1
      REWIND IU2
      REWIND IU3
      REWIND IU4
      DO 9 KK=1,NOP
      KA=(KK-1)*NROW
      DO 4 IXBLK1=1,NBLSYM
      CALL BLCKIN (A,IU2,I1,I2,1,121)
      K1=(IXBLK1-1)*NPSYM+2
      IF (NM1.LT.K1) GO TO 3
      J2=0
      DO 2 K=K1,NM1
      IF (J2.LT.NPSYM) J2=J2+1
      IPK=IP(K+KA)
      DO 1 J=1,J2
      TEMP=A(K,J)
      A(K,J)=A(IPK,J)
      A(IPK,J)=TEMP
1     CONTINUE
2     CONTINUE
3     CONTINUE
      CALL BLCKOT (A,IU3,I1,I2,1,122)
4     CONTINUE
      DO 5 IXBLK1=1,NBLSYM
      BACKSPACE IU3
      IF (IXBLK1.NE.1) BACKSPACE IU3
      CALL BLCKIN (A,IU3,I1,I2,1,123)
      CALL BLCKOT (A,IU4,I1,I2,1,124)
5     CONTINUE
      DO 6 I=1,NROW
      IX(I+KA)=I
6     CONTINUE
      DO 7 I=1,NROW
      IPI=IP(I+KA)
      IXT=IX(I+KA)
      IX(I+KA)=IX(IPI+KA)
      IX(IPI+KA)=IXT
7     CONTINUE
      IF (NOP.EQ.1) GO TO 9
      NB1=NBLSYM-1
C     SKIP NB1 LOGICAL RECORDS FORWARD
      DO 8 IXBLK1=1,NB1
      CALL BLCKIN (A,IU3,I1,I2,1,125)
8     CONTINUE
9     CONTINUE
      REWIND IU2
      REWIND IU3
      REWIND IU4
      RETURN
      END
```

MOVE

PURPOSE

To rotate and translate a previously defined structure, either moving original segments and patches or leaving the original fixed and producing new segments and patches.

METHOD

The formal parameters ROX, ROY, ROZ are the angles of rotation about the x, y, and z axes, respectively, and XS, YS, ZS are the translation distances in the x, y, and z directions. Angles are in radians, and a positive angle represents a right-hand rotation. The structure is first rotated about the x axis by ROX, then about the y axis by ROY, then about the z axis by ROZ, and finally translated by XS, YS, ZS. These operations transform a point with coordinates x, y, z to x', y', z', where

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix}$$

where

$T_{11} = \cos\phi \cos\theta$

$T_{12} = \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi$

$T_{13} = \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi$

$T_{21} = \sin\phi \cos\theta$

$T_{22} = \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi$

$T_{23} = \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi$

$T_{31} = -\sin\theta$

$T_{32} = \cos\theta \sin\psi$

$T_{33} = \cos\theta \cos\psi$

with

ψ = ROX

θ = ROY

ϕ = ROZ

X_s = XS

Y_s = YS

Z_s = ZS

This transformation is applied to those wire segments from segment number i_s to the last defined segment in COMMON/DATA/. Thus, if i_s is greater than 1, the segments from 1 to $i_s - 1$ are unaffected. All patches are transformed.

NRPT is the structure repetition factor. If NRPT is zero, the transformed segment and patch coordinates overwrite the original coordinates so that the structure is moved with nothing left in the original location. If NRPT is greater than zero, the transformed coordinates are written on the ends of the arrays in COMMON/DATA/ and the process repeated NRPT times so that NRPT new structures are formed, each shifted from the previous one by the specified transformation, while the original structure is unchanged.

CODING

MO18	Adjust symmetry flag if structure is rotated about the x or y axis. If the ground plane flag is also set on the GE card, symmetry will not be used in the solution.
MO19 - MO33	Compute transformation matrix.
MO37 - MO61	Transform segment coordinates.
MO63 - MO93	Transform patch coordinates.
MO94 - MO97	Set parameters to no-symmetry condition if NRPT > 0 or IX > 1.

SYMBOL DICTIONARY

ABS	= external routine (absolute value)
COS	= external routine (cosine)
CPH	= cos ϕ
CPS	= cos Ψ
CTH	= cos θ
IR	= DO loop index, array index for original patch
ISEGNO	= external routine (searches segment tag numbers)
ITGI	= increment applied to segment tag numbers as segments are transformed
ITS	= i_s is the first occurring segment in COMMON/DATA/ with tag ITS
IX	= i_s
I1	= lower DO loop limit for I (initially I1 = i_s)
K	= increment to segment number for transformed segment
KR	= array index for new patch

LDI = LD + 1
NRP = upper DO loop limit for IR
NRPT = repetition factor
ROX = Ψ (radians)
ROY = θ
ROZ = ϕ
SIN = external routine (sine)
SPH = $\sin \phi$
SPS = $\sin \Psi$
STH = $\sin \theta$
T1X, T1Y, T1Z = arrays containing components of $\hat{t}_1$ for patches
T2X, T2Y, T2Z = arrays containing components of $\hat{t}_2$ for patches
XI = old x coordinate
XS = x_s
XX = T_{11}
XY = T_{12}
XZ = T_{13}
X2(I) = x coordinate of end 2 of segment I
YI = old y coordinate
YS = y_s
YX = T_{21}
YY = T_{22}
YZ = T_{23}
Y2(I) = y coordinate of end 2 of segment I
ZI = old Z coordinate
ZS = Z_s
ZX = T_{31}
ZY = T_{32}
ZZ = T_{33}
Z2(I) = Z coordinate of end 2 of segment I

```
      SUBROUTINE MOVE (ROX,ROY,ROZ,XS,YS,ZS,ITS,NRPT,ITGI)
C
C     SUBROUTINE MOVE MOVES THE STRUCTURE WITH RESPECT TO ITS
C     COORDINATE SYSTEM OR REPRODUCES STRUCTURE IN NEW POSITIONS.
C     STRUCTURE IS ROTATED ABOUT X,Y,Z AXES BY ROX,ROY,ROZ
C     RESPECTIVELY, THEN SHIFTED BY XS,YS,ZS
C
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), X2(1), Y
     12(1), Z2(1)
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      IF (ABS(ROX)+ABS(ROY).GT.1.E-10) IPSYM=IPSYM*3
      SPS=SIN(ROX)
      CPS=COS(ROX)
      STH=SIN(ROY)
      CTH=COS(ROY)
      SPH=SIN(ROZ)
      CPH=COS(ROZ)
      XX=CPH*CTH
      XY=CPH*STH*SPS-SPH*CPS
      XZ=CPH*STH*CPS+SPH*SPS
      YX=SPH*CTH
      YY=SPH*STH*SPS+CPH*CPS
      YZ=SPH*STH*CPS-CPH*SPS
      ZX=-STH
      ZY=CTH*SPS
      ZZ=CTH*CPS
      NRP=NRPT
      IF (NRPT.EQ.0) NRP=1
      IF (N.LT.N2) GO TO 3
      I1=ISEGNO(ITS,1)
      IF (I1.LT.N2) I1=N2
      IX=I1
      K=N
      IF (NRPT.EQ.0) K=I1-1
      DO 2 IR=1,NRP
      DO 1 I=I1,N
      K=K+1
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      X(K)=XI*XX+YI*XY+ZI*XZ+XS
      Y(K)=XI*YX+YI*YY+ZI*YZ+YS
      Z(K)=XI*ZX+YI*ZY+ZI*ZZ+ZS
      XI=X2(I)
      YI=Y2(I)
      ZI=Z2(I)
      X2(K)=XI*XX+YI*XY+ZI*XZ+XS
      Y2(K)=XI*YX+YI*YY+ZI*YZ+YS
      Z2(K)=XI*ZX+YI*ZY+ZI*ZZ+ZS
      BI(K)=BI(I)
      ITAG(K)=ITAG(I)+ITGI
1     CONTINUE
      I1=N+1
      N=K
2     CONTINUE
3     IF (M.LT.M2) GO TO 6
      I1=M2
      K=M
```

```
      LDI=LD+1
      IF (NRPT.EQ.0) K=M1
      DO 5 II=1,NRP
      DO 4 I=I1,M
      K=K+1
      IR=LDI-I
      KR=LDI-K
      XI=X(IR)
      YI=Y(IR)
      ZI=Z(IR)
      X(KR)=XI*XX+YI*XY+ZI*XZ+XS
      Y(KR)=XI*YX+YI*YY+ZI*YZ+YS
      Z(KR)=XI*ZX+YI*ZY+ZI*ZZ+ZS
      XI=T1X(IR)
      YI=T1Y(IR)
      ZI=T1Z(IR)
      T1X(KR)=XI*XX+YI*XY+ZI*XZ
      T1Y(KR)=XI*YX+YI*YY+ZI*YZ
      T1Z(KR)=XI*ZX+YI*ZY+ZI*ZZ
      XI=T2X(IR)
      YI=T2Y(IR)
      ZI=T2Z(IR)
      T2X(KR)=XI*XX+YI*XY+ZI*XZ
      T2Y(KR)=XI*YX+YI*YY+ZI*YZ
      T2Z(KR)=XI*ZX+YI*ZY+ZI*ZZ
      SALP(KR)=SALP(IR)
4     BI(KR)=BI(IR)
      I1=M+1
5     M=K
6     IF ((NRPT.EQ.0).AND.(IX.EQ.1)) RETURN
      NP=N
      MP=M
      IPSYM=0
      RETURN
      END
```

NEFLD

PURPOSE

To compute the near electric field due to currents induced on a structure.

CODING

NE30 - NE93	Near E field due to currents on segments is computed.
NE30 - NE41	Each segment is checked to determine whether the field observation point (XOB, YOB, ZOB) falls within the segment volume. If it does, AX is set to the radius of that segment. AX is then sent to routine EFLD as the radius of the observation segment. If (XOB, YOB, ZOB) is on the axis of a segment at its center, the field calculation with AX set to the segment radius is the same as that used in filling the matrix.
NE42 - NE93	Loop computing the field contribution of each segment.
NE43 - NE50	Parameters of source segment are stored in COMMON/DATAJ/.
NE51 - NE85	When the extended thin wire approximation is used, IND1 is set to 0 if end 1 of segment I is connected to a single parallel segment of the same radius, 1 if it is a free end, and 2 if it connects to a multiple junction, a bend, or a segment of different radius. IND2 is the same for end 2. If IND1 or IND2 is 2, the extended thin wire approximation will not be used for that end.
NE87	EFLD stores the electric fields due to constant, sin ks, and cos ks currents in COMMON/DATAJ/.
NE88 - NE93	The field components are multiplied by the coefficients of the constant, sin ks, and cos ks components of the total segment current, and the field is summed.
NE95 - NE117	Near field due to patch currents is computed.

SYMBOL DICTIONARY

ACX = constant component of segment current at NE88; $\hat{t}_1$ component of patch current at NE110

AX = segment radius when the field evaluation point falls within a segment volume

B = source segment radius

BCX	= sin ks component of segment current at NE89; $\hat{t}_2$ component of patch current at NE111
CCX	= cos ks component of segment current at NE90
EX EY EZ	= x, y, and z components of total electric field
EXC EYC EZC	= E field due to a cos ks current on a segment
EXK EYK EZK	= E field due to a constant current at NE87; E field due to the $\hat{t}_1$ component of patch current at NE114
EXS EYX EZS	= E field due to a sin ks current at NE87; E field due to the $\hat{t}_2$ component of patch current at NE114
IP	= loop index for direct and reflected field (1, 2, respectively)
T1X T1Y T1Z	= arrays for $\hat{t}_1$
T1XJ T1YJ T1ZJ	= $\hat{t}_1$ for source patch
T2X T2Y T2Z	= arrays for $\hat{t}_2$
T2XJ T2YJ T2ZJ	= $\hat{t}_2$ for source path
XI	= cosine of the angle between segment I and the segment connected to its end
XOB YOB ZOB	= field evaluation point
ZP	= coordinates of the field evaluation point, z or ρ^2, in a cylindrical coordinate system centered on the source segment

CONSTANTS

0.5001 = fraction of segment length used to test whether the field evaluation point falls within a segment

0.9 = fraction of segment radius used to test whether the field evaluation point falls within a segment

0.999999 = minimum XI for extended thin wire kernel (maximum angle = 0.08 degree)

```
      SUBROUTINE NEFLD (XOB,YOB,ZOB,EX,EY,EZ)
C
C     NEFLD COMPUTES THE NEAR FIELD AT SPECIFIED POINTS IN SPACE AFTER
C     THE STRUCTURE CURRENTS HAVE BEEN COMPUTED.
C
      COMPLEX EX,EY,EZ,CUR,ACX,BCX,CCX,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,E
     1ZC,ZRATI,ZRATI2,T1,FRATI
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),CII(300
     1),CUR(900)
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      DIMENSION CAB(1), SAB(1), T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1),
     1T2Z(1)
      EQUIVALENCE (CAB,ALP), (SAB,BET)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG)
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y
     1J,IND1), (T2ZJ,IND2)
      EX=(0.,0.)
      EY=(0.,0.)
      EZ=(0.,0.)
      AX=0.
      IF (N.EQ.0) GO TO 20
      DO 1 I=1,N
      XJ=XOB-X(I)
      YJ=YOB-Y(I)
      ZJ=ZOB-Z(I)
      ZP=CAB(I)*XJ+SAB(I)*YJ+SALP(I)*ZJ
      IF (ABS(ZP).GT.0.5001*SI(I)) GO TO 1
      ZP=XJ*XJ+YJ*YJ+ZJ*ZJ-ZP*ZP
      XJ=BI(I)
      IF (ZP.GT.0.9*XJ*XJ) GO TO 1
      AX=XJ
      GO TO 2
1     CONTINUE
2     DO 19 I=1,N
      S=SI(I)
      B=BI(I)
      XJ=X(I)
      YJ=Y(I)
      ZJ=Z(I)
      CABJ=CAB(I)
      SABJ=SAB(I)
      SALPJ=SALP(I)
      IF (IEXK.EQ.0) GO TO 18
      IPR=ICON1(I)
      IF (IPR) 3,8,4
3     IPR=-IPR
      IF (-ICON1(IPR).NE.I) GO TO 9
      GO TO 6
4     IF (IPR.NE.I) GO TO 5
      IF (CABJ*CABJ+SABJ*SABJ.GT.1.E-8) GO TO 9
      GO TO 7
5     IF (ICON2(IPR).NE.I) GO TO 9
6     XI=ABS(CABJ*CAB(IPR)+SABJ*SAB(IPR)+SALPJ*SALP(IPR))
      IF (XI.LT.0.999999) GO TO 9
      IF (ABS(BI(IPR)/B-1.).GT.1.E-6) GO TO 9
7     IND1=0
```

```
      GO TO 10
    8 IND1=1
      GO TO 10
    9 IND1=2
   10 IPR=ICON2(I)
      IF (IPR) 11,16,12
   11 IPR=-IPR
      IF (-ICON2(IPR).NE.I) GO TO 17
      GO TO 14
   12 IF (IPR.NE.I) GO TO 13
      IF (CABJ*CABJ+SABJ*SABJ.GT.1.E-8) GO TO 17
      GO TO 15
   13 IF (ICON1(IPR).NE.I) GO TO 17
   14 XI=ABS(CABJ*CAB(IPR)+SABJ*SAB(IPR)+SALPJ*SALP(IPR))
      IF (XI.LT.0.999999) GO TO 17
      IF (ABS(BI(IPR)/B-1.).GT.1.E-6) GO TO 17
   15 IND2=0
      GO TO 18
   16 IND2=1
      GO TO 18
   17 IND2=2
   18 CONTINUE
      CALL EFLD (XOB,YOB,ZOB,AX,1)
      ACX=CMPLX(AIR(I),AII(I))
      BCX=CMPLX(BIR(I),BII(I))
      CCX=CMPLX(CIR(I),CII(I))
      EX=EX+EXK*ACX+EXS*BCX+EXC*CCX
      EY=EY+EYK*ACX+EYS*BCX+EYC*CCX
   19 EZ=EZ+EZK*ACX+EZS*BCX+EZC*CCX
      IF (M.EQ.0) RETURN
   20 JC=N
      JL=LD+1
      DO 21 I=1,M
      JL=JL-1
      S=BI(JL)
      XJ=X(JL)
      YJ=Y(JL)
      ZJ=Z(JL)
      T1XJ=T1X(JL)
      T1YJ=T1Y(JL)
      T1ZJ=T1Z(JL)
      T2XJ=T2X(JL)
      T2YJ=T2Y(JL)
      T2ZJ=T2Z(JL)
      JC=JC+3
      ACX=T1XJ*CUR(JC-2)+T1YJ*CUR(JC-1)+T1ZJ*CUR(JC)
      BCX=T2XJ*CUR(JC-2)+T2YJ*CUR(JC-1)+T2ZJ*CUR(JC)
      DO 21 IP=1,KSYMP
      IPGND=IP
      CALL UNERE (XOB,YOB,ZOB)
      EX=EX+ACX*EXK+BCX*EXS
      EY=EY+ACX*EYK+BCX*EYS
   21 EZ=EZ+ACX*EZK+BCX*EZS
      RETURN
      END
```

NETWK

PURPOSE

To solve for the voltages and currents at the ports of non-radiating networks that are part of the antenna. This routine also is involved in the solution for current when there are no non-radiating networks, and computes the relative driving point matrix asymmetry when this option is requested.

METHOD

Driving Point Matrix Asymmetry (NT32 to NT84):

To satisfy physical reciprocity, the elements of the inverse of the interaction matrix should satisfy the condition

$$G^{-1}_{ij}/\Delta_j = G^{-1}_{ji}/\Delta_i \qquad i, j = 1, \ldots, n ,$$

where Δ_i = length of segment i. This condition is not satisfied exactly, except on special structures, since the terms computed are not true reactions. The relative asymmetry of a matrix element is defined as

$$A = \left| \frac{\left(G^{-1}_{ij}/\Delta_j - G^{-1}_{ji}/\Delta_i\right)}{(G^{-1}_{ij}/\Delta_j)} \right| .$$

The code from NT32 to NT84 computes the relative asymmetries of matrix elements for i and j of all driving point segments: either voltage source driving points or network connection points. The maximum relative asymmetry is located, and the rms relative asymmetry of all elements used is computed.

LOCAL CODING STRUCTURE

NT32 - NT44 Determine numbers of segments that are network connection points.

NT46 - NT54 Determine numbers of segments that are voltage source driving points. Indices of segments with network connections or voltage sources are stored in array IPNT with no duplication of numbers.

NT59 - NT69 Compute $G^{-1}_{k\ell}/\Delta_\ell$ for k,ℓ = all segment numbers in IPNT.

NT70 - NT84 Compute relative asymmetries of elements computed above, search for maximum and compute rms asymmetry.

LOCAL SYMBOL DICTIONARY

ASA = sum of squares of relative asymmetries and rms value

ASM = Δ_{ISC1} before NT70; maximum relative asymmetry after NT69

CMN(J,I) = $G^{-1}_{k\ell}/\Delta_\ell$; k = IPNT(J), ℓ = IPNT(I)

CUR = temporary storage of $G^{-1}_{\ell k}/\Delta_k$

IPNT = array of driving point segment indices

IROW1 = number of entries in IPNT

ISC1 = temporary storage of segment index

MASYM = flag; if non-zero, matrix asymmetry is computed

NTEQ = row index of element having maximum asymmetry

NTSC = column index of element having maximum asymmetry

PWR = relative matrix asymmetry

RHS = vector for matrix solution used in obtaining $G^{-1}_{k\ell}$

<u>Non-radiating Network Solution (NT89 to NT262):</u>

The solution method when non-radiating networks are present is discussed in Part I.

Data for non-radiating networks is passed through the COMMON/NETCX/ where

ISEG1(I) = number of the segment to which end 1 of I^{th} two-port network is connected

ISEG2(I) = number of segment to which end 2 of I^{th} two-port network is connected

NONET = number of two-port networks for which data is given

Network parameters are contained in the arrays X11R, X11I, X12R, X12I, X22R, and X22I, and the type of network is determined by NTYP:

If NTYP is 1 -- the network parameters are the short-circuit admittance parameters of the network:

X11R, X11I = real and imaginary parts of Y_{11}

X12R, X12I = real and imaginary parts of $Y_{12} = Y_{21}$

X22R, X22I = real and imaginary parts of Y_{22}

If NTYP is 2 or 3 -- the network is a transmission line:

X11R = characteristic impedance of transmission line

X11I = length of transmission line in meters

X12R = real part of shunt admittance on end 1 of line

X12I = imaginary part of shunt admittance on end 1 of line

X22R = real part of shunt admittance on end 2 of line

X22I = imaginary part of shunt admittance on end 2 of line

If NTYP is 2 -- the transmission line runs straight between the segments with respect to the segment reference directions.

If NTYP is 3 -- the transmission line is twisted as shown in figure 8.

The short circuit admittance parameters of the transmission line, Y_{11}, Y_{12}, and Y_{22}, are computed from NT110 to NT120 in the code. When NTYP is 3, the sign of Y_{12} is reversed.

The code from NT99 to NT194 forms a loop that for each network: computes the network parameters Y_{11}, Y_{12} and Y_{22}; sorts the segment indices involved; and adds the parameters Y_{11}, Y_{12}, and Y_{22} to the appropriate network equations. The sorting procedure for the connection of end 1 of the network is described in figure 9. Decision 1 is made in the code from NT121 to NT126, decision 2 from NT128 to NT133, and decision 3 from NT138 to NT143. Segments having network connections only are assigned equation rows in the array CMN starting from the top in the order that the segments are encountered. Segments with both network and voltage source connections are assigned equation rows in CMN starting at the bottom and proceeding up. The former are eventually solved for the unknown gap voltages, while the latter are used to obtain source input admittances after the structure currents have been computed. The code from NT148 to NT174 assigns equation numbers for the connection of end 2 of the networks and sets IROW2 and ISC2.

The network short circuit parameters are added to the network equations from NT182 to NT193. The coefficient matrix is transposed in filling the CMN array, since the matrix solution routines operate on a transposed system. Hence, the first index should be considered the column number and the second index the row number. If a segment NSEG1 does not have a voltage source connected, the parameters Y_{11} and Y_{12} are added to column IROW1 at rows IROW1 and IROW2, respectively. IROW2 may be either (1) in the upper rows as part of the equations for the unknown gap voltages, or (2) if a voltage source is connected to segment NSEG2, in the lower rows for later determination of the source current. If a voltage source is connected to segment NSEG1, the

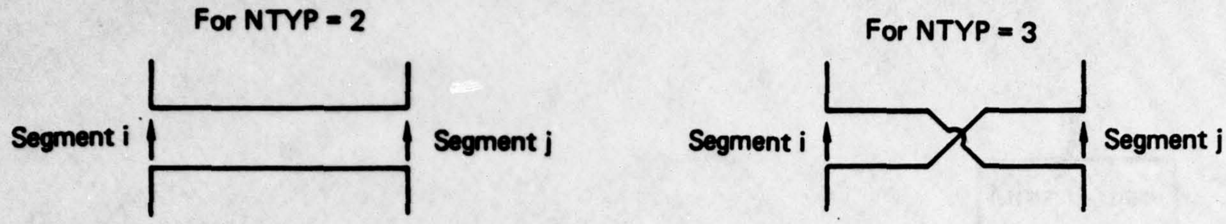


Figure 8. Options for Transmission Line Connection.

coefficients Y_{11} and Y_{12} are multiplied by the known source voltage and added to the right-hand side of the network equation in the rows IROW1 and IROW2. The parameters Y_{12} and Y_{22} are added to the equations in a similar manner.

The loop from NT199 to NT208 computes the elements of the inverse matrix G^{-1}_{mm} and adds them to the network equations. The network matrix is then factored at NT213. The code from NT218 to NT225 computes B_i = RHS(I), where

$$B_i = \sum_{j=1}^{N} G^{-1}_{ij} E'_j \quad i = 1, \ldots N ,$$

with $(-E'_j)$ being the known applied field on segment j, not including unknown voltage drops at network ports. Those elements B_i for segments in the network equations are then added to the right-hand side of the network equations. At NT229 the network equations are solved for the excitation fields due to voltage drops at the network ports. The negatives of these fields are added to the excitation vector at NT234 to NT236, completing the definition of the excitation vector E_j. The structure equations are then solved for the induced currents.

$$I_j = \sum_{j=1}^{N} G^{-1}_{ij} E_j .$$

From NT241 to NT261, the voltage, current, admittance, and power seen looking into the structure at each network port are printed. This current does not include current through any voltage sources that are connected to the port.

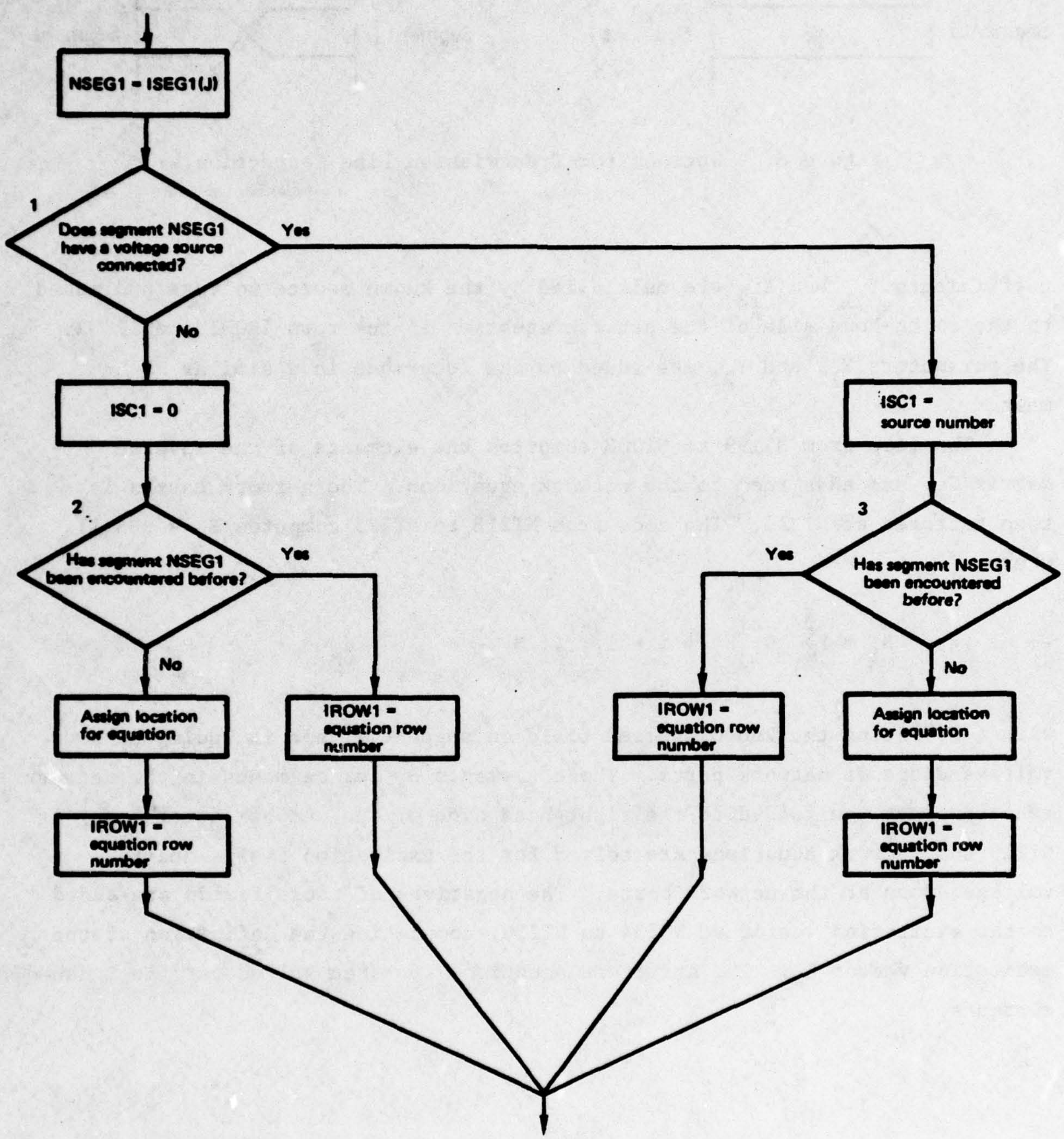


Figure 9. Sorting Procedure for Segments Having Network Connections.

The code from NT269 to NT294 computes and prints the voltage, current, admittance, and power seen by each voltage source looking into the structure and parallel connected network port, if a network is present.

After the network equations have once been set up, they can be solved for various incident fields by entering the code at NT218. If the location of voltage sources is changed, however, the equations must be recomputed.

If a structure has no non-radiating networks, the currents are computed at NT266.

SYMBOL DICTIONARY

ASA	= sum of squares of relative matrix asymmetries and rms value
ASM	= segment length and maximum relative matrix asymmetry
CABS	= external routine (magnitude of complex number)
CM	= array of matrix elements G_{ij}
CMN	= array for network equation coefficients
CMPLX	= external routine (forms complex number)
CONJG	= external routine (conjugate)
COS	= external routine (cosine)
CUR	= current
EINC	= excitation vector
FACTR	= external routine (Gauss-Doolittle matrix factoring)
FLOAT	= external routine (integer to real conversion)
I	= DO loop index
IP	= array of positioning data from factoring of CM
IPNT	= array of positioning data from factoring of CMN
IROW1	= matrix element index
IROW2	= matrix element index
ISANT	= array of segment numbers for voltage source connection
ISC1	= segment location in array ISANT
ISC2	= segment location in array ISANT
ISEG1	= number of segment to which port 1 of network is connected
ISEG2	= number of segment to which port 2 is connected
IX	= array of positioning data from factoring of CM
J	= DO loop index
MASYM	= flag to request matrix asymmetry calculation
NCOL	= number of columns in CM
NDIMN	= array dimension of CMN

NDIMNP = NDIMN + 1

NONET = number of networks

NOP = N/NP

NPRINT = flag to control printing

NROW = number of rows in CM

NSANT = number of voltage sources

NSEG1 = array of segments to which port 1 of a network connects

NSEG2 = array of segments to which port 2 of a network connects

NTEQA(I) = segment number associated with I^{th} network equation

NTSC = number of network-voltage source equations

NTSCA(I) = segment number associated with I^{th} network-voltage source equation

NTSOL = flag to indicate network equations do not need to be recomputed

NTYP(I) = type of I^{th} network

PIN = total input power from sources

PNLS = power lost in networks

PWR = power

REAL = external routine (real part of complex number)

RHNT = vector for right-hand side of network equations

RHNX = component of RHNT due to Y_{11}, Y_{12}, Y_{22} terms

RHS = vector for right-hand side of structure interaction equation

SIN = external routine (sine)

SOLVE = external routine (Gauss-Doolittle solution)

SOLVES = external routine (Gauss-Doolittle solution of CM matrix)

SQRT = external routine (square root)

TP = 2π

VLT = voltage

VSANT(I) = voltage of source on segment NSANT(I)

VSRC(I) = voltage of source on I^{th} segment in network-voltage source equations

X11I, X11R, X12I, X12R, X22I, X22R = network or transmission line specification parameters

YMIT = admittance

Y11I = imaginary part of Y_{11}

Y11R = real part of Y_{11}

Y12I = imaginary part of Y_{12}

Y12R = real part of Y_{12}

Y22I = imaginary part of Y_{22}

Y22R = real part of Y_{22}

ZPED = impedance

CONSTANTS

6.283185308 = 2π

30 = row and column dimensions of CMN

31 = (row and column dimensions of CMN) + 1

```
      SUBROUTINE NETWK (CM,CMB,CMC,CMD,IP,EINC)
C
C     SUBROUTINE NETWK SOLVES FOR STRUCTURE CURRENTS FOR A GIVEN
C     EXCITATION INCLUDING THE EFFECT OF NON-RADIATING NETWORKS IF
C     PRESENT.
C
      COMPLEX CMN,RHNT,YMIT,RHS,ZPED,EINC,VSANT,VLT,CUR,VSRC,RHNX,VQD,VQ
     1DS,CUX,CM,CMB,CMC,CMD
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),CII(300
     1),CUR(900)
      COMMON /VSORC/ VQD(30),VSANT(30),VQDS(30),IVQD(30),ISANT(30),IQDS(
     130),NVQD,NSANT,NQDS
      COMMON /NETCX/ ZPED,PIN,PNLS,NEQ,NPEQ,NEQ2,NONET,NTSOL,NPRINT,MASY
     1M,ISEG1(30),ISEG2(30),X11R(30),X11I(30),X12R(30),X12I(30),X22R(30)
     2,X22I(30),NTYP(30)
      DIMENSION EINC(1), IP(1)
      DIMENSION CMN(30,30), RHNT(30), IPNT(30), NTEQA(30), NTSCA(30), RH
     1S(900), VSRC(10), RHNX(30)
      DATA NDIMN,NDIMNP/30,31/,TP/6.283185308/
      PIN=0.
      PNLS=0.
      NEQT=NEQ+NEQ2
      IF (NTSOL.NE.0) GO TO 42
      NOP=NEQ/NPEQ
      IF (MASYM.EQ.0) GO TO 14
C
C     COMPUTE RELATIVE MATRIX ASYMMETRY
C
      IROW1=0
      IF (NONET.EQ.0) GO TO 5
      DO 4 I=1,NONET
      NSEG1=ISEG1(I)
      DO 3 ISC1=1,2
      IF (IROW1.EQ.0) GO TO 2
      DO 1 J=1,IROW1
      IF (NSEG1.EQ.IPNT(J)) GO TO 3
1     CONTINUE
2     IROW1=IROW1+1
      IPNT(IROW1)=NSEG1
3     NSEG1=ISEG2(I)
4     CONTINUE
5     IF (NSANT.EQ.0) GO TO 9
      DO 8 I=1,NSANT
      NSEG1=ISANT(I)
      IF (IROW1.EQ.0) GO TO 7
      DO 6 J=1,IROW1
      IF (NSEG1.EQ.IPNT(J)) GO TO 8
6     CONTINUE
7     IROW1=IROW1+1
      IPNT(IROW1)=NSEG1
8     CONTINUE
9     IF (IROW1.LT.NDIMNP) GO TO 10
      PRINT 59
      STOP
10    IF (IROW1.LT.2) GO TO 14
      DO 12 I=1,IROW1
      ISC1=IPNT(I)
      ASM=SI(ISC1)
      DO 11 J=1,NEQT
11    RHS(J)=(0.,0.)
      RHS(ISC1)=(1.,0.)
```

```
      CALL SOLGF (CM,CMB,CMC,CMD,RHS,IP,NP,N1,N,MP,M1,M,NEQ,NEQ2)
      CALL CABC (RHS)
      DO 12 J=1,IROW1
      ISC1=IPNT(J)
   12 CMN(J,I)=RHS(ISC1)/ASM
      ASM=0.
      ASA=0.
      DO 13 I=2,IROW1
      ISC1=I-1
      DO 13 J=1,ISC1
      CUX=CMN(I,J)
      PWR=CABS((CUX-CMN(J,I))/CUX)
      ASA=ASA+PWR*PWR
      IF (PWR.LT.ASM) GO TO 13
      ASM=PWR
      NTEQ=IPNT(I)
      NTSC=IPNT(J)
   13 CONTINUE
      ASA=SQRT(ASA*2./FLOAT(IROW1*(IROW1-1)))
      PRINT 58, ASM,NTEQ,NTSC,ASA
   14 IF (NONET.EQ.0) GO TO 48
C
C     SOLUTION OF NETWORK EQUATIONS
C
      DO 15 I=1,NDIMN
      RHNX(I)=(0.,0.)
      DO 15 J=1,NDIMN
   15 CMN(I,J)=(0.,0.)
      NTEQ=0
      NTSC=0
C
C     SORT NETWORK AND SOURCE DATA AND ASSIGN EQUATION NUMBERS TO
C     SEGMENTS.
C
      DO 38 J=1,NONET
      NSEG1=ISEG1(J)
      NSEG2=ISEG2(J)
      IF (NTYP(J).GT.1) GO TO 16
      Y11R=X11R(J)
      Y11I=X11I(J)
      Y12R=X12R(J)
      Y12I=X12I(J)
      Y22R=X22R(J)
      Y22I=X22I(J)
      GO TO 17
   16 Y22R=TP*X11I(J)/WLAM
      Y12R=0.
      Y12I=1./(X11R(J)*SIN(Y22R))
      Y11R=X12R(J)
      Y11I=-Y12I*COS(Y22R)
      Y22R=X22R(J)
      Y22I=Y11I+X22I(J)
      Y11I=Y11I+X12I(J)
      IF (NTYP(J).EQ.2) GO TO 17
      Y12R=-Y12R
      Y12I=-Y12I
   17 IF (NSANT.EQ.0) GO TO 19
      DO 18 I=1,NSANT
      IF (NSEG1.NE.ISANT(I)) GO TO 18
      ISC1=I
      GO TO 22
   18 CONTINUE
   19 ISC1=0
      IF (NTEQ.EQ.0) GO TO 21
```

```
      DO 20 I=1,NTEQ
      IF (NSEG1.NE.NTEQA(I)) GO TO 20
      IROW1=I
      GO TO 25
   20 CONTINUE
   21 NTEQ=NTEQ+1
      IROW1=NTEQ
      NTEQA(NTEQ)=NSEG1
      GO TO 25
   22 IF (NTSC.EQ.0) GO TO 24
      DO 23 I=1,NTSC
      IF (NSEG1.NE.NTSCA(I)) GO TO 23
      IROW1=NDIMNP-I
      GO TO 25
   23 CONTINUE
   24 NTSC=NTSC+1
      IROW1=NDIMNP-NTSC
      NTSCA(NTSC)=NSEG1
      VSRC(NTSC)=VSANT(ISC1)
   25 IF (NSANT.EQ.0) GO TO 27
      DO 26 I=1,NSANT
      IF (NSEG2.NE.ISANT(I)) GO TO 26
      ISC2=I
      GO TO 30
   26 CONTINUE
   27 ISC2=0
      IF (NTEQ.EQ.0) GO TO 29
      DO 28 I=1,NTEQ
      IF (NSEG2.NE.NTEQA(I)) GO TO 28
      IROW2=I
      GO TO 33
   28 CONTINUE
   29 NTEQ=NTEQ+1
      IROW2=NTEQ
      NTEQA(NTEQ)=NSEG2
      GO TO 33
   30 IF (NTSC.EQ.0) GO TO 32
      DO 31 I=1,NTSC
      IF (NSEG2.NE.NTSCA(I)) GO TO 31
      IROW2=NDIMNP-I
      GO TO 33
   31 CONTINUE
   32 NTSC=NTSC+1
      IROW2=NDIMNP-NTSC
      NTSCA(NTSC)=NSEG2
      VSRC(NTSC)=VSANT(ISC2)
   33 IF (NTSC+NTEQ.LT.NDIMNP) GO TO 34
      PRINT 59
      STOP
C
C     FILL NETWORK EQUATION MATRIX AND RIGHT HAND SIDE VECTOR WITH
C     NETWORK SHORT-CIRCUIT ADMITTANCE MATRIX COEFFICIENTS.
C
   34 IF (ISC1.NE.0) GO TO 35
      CMN(IROW1,IROW1)=CMN(IROW1,IROW1)-CMPLX(Y11R,Y11I)*SI(NSEG1)
      CMN(IROW1,IROW2)=CMN(IROW1,IROW2)-CMPLX(Y12R,Y12I)*SI(NSEG1)
      GO TO 36
   35 RHNX(IROW1)=RHNX(IROW1)+CMPLX(Y11R,Y11I)*VSANT(ISC1)/WLAM
      RHNX(IROW2)=RHNX(IROW2)+CMPLX(Y12R,Y12I)*VSANT(ISC1)/WLAM
   36 IF (ISC2.NE.0) GO TO 37
      CMN(IROW2,IROW2)=CMN(IROW2,IROW2)-CMPLX(Y22R,Y22I)*SI(NSEG2)
      CMN(IROW2,IROW1)=CMN(IROW2,IROW1)-CMPLX(Y12R,Y12I)*SI(NSEG2)
      GO TO 38
   37 RHNX(IROW1)=RHNX(IROW1)+CMPLX(Y12R,Y12I)*VSANT(ISC2)/WLAM
```

```
      RHNX(IROW2)=RHNX(IROW2)+CMPLX(Y22R,Y22I)*VSANT(ISC2)/WLAM          NT 193
   38 CONTINUE                                                          NT 194
C                                                                       NT 195
C     ADD INTERACTION MATRIX ADMITTANCE ELEMENTS TO NETWORK EQUATION    NT 196
C     MATRIX                                                            NT 197
C                                                                       NT 198
      DO 41 I=1,NTEQ                                                    NT 199
      DO 39 J=1,NEQT                                                    NT 200
   39 RHS(J)=(0.,0.)                                                    NT 201
      IROW1=NTEQA(I)                                                    NT 202
      RHS(IROW1)=(1.,0.)                                                NT 203
      CALL SOLGF (CM,CMB,CMC,CMD,RHS,IP,NP,N1,N,MP,M1,M,NEQ,NEQ2)       NT 204
      CALL CABC (RHS)                                                   NT 205
      DO 40 J=1,NTEQ                                                    NT 206
      IROW1=NTEQA(J)                                                    NT 207
   40 CMN(I,J)=CMN(I,J)+RHS(IROW1)                                      NT 208
   41 CONTINUE                                                          NT 209
C                                                                       NT 210
C     FACTOR NETWORK EQUATION MATRIX                                    NT 211
C                                                                       NT 212
      CALL FACTR (NTEQ,CMN,IPNT,NDIMN)                                  NT 213
C                                                                       NT 214
C     ADD TO NETWORK EQUATION RIGHT HAND SIDE THE TERMS DUE TO ELEMENT  NT 215
C     INTERACTIONS                                                      NT 216
C                                                                       NT 217
   42 IF (NONET.EQ.0) GO TO 48                                          NT 218
      DO 43 I=1,NEQT                                                    NT 219
   43 RHS(I)=EINC(I)                                                    NT 220
      CALL SOLGF (CM,CMB,CMC,CMD,RHS,IP,NP,N1,N,MP,M1,M,NEQ,NEQ2)       NT 221
      CALL CABC (RHS)                                                   NT 222
      DO 44 I=1,NTEQ                                                    NT 223
      IROW1=NTEQA(I)                                                    NT 224
   44 RHNT(I)=RHNX(I)+RHS(IROW1)                                        NT 225
C                                                                       NT 226
C     SOLVE NETWORK EQUATIONS                                           NT 227
C                                                                       NT 228
      CALL SOLVE (NTEQ,CMN,IPNT,RHNT,NDIMN)                             NT 229
C                                                                       NT 230
C     ADD FIELDS DUE TO NETWORK VOLTAGES TO ELECTRIC FIELDS APPLIED TO  NT 231
C     STRUCTURE AND SOLVE FOR INDUCED CURRENT                           NT 232
C                                                                       NT 233
      DO 45 I=1,NTEQ                                                    NT 234
      IROW1=NTEQA(I)                                                    NT 235
   45 EINC(IROW1)=EINC(IROW1)-RHNT(I)                                   NT 236
      CALL SOLGF (CM,CMB,CMC,CMD,EINC,IP,NP,N1,N,MP,M1,M,NEQ,NEQ2)      NT 237
      CALL CABC (EINC)                                                  NT 238
      IF (NPRINT.EQ.0) PRINT 61                                         NT 239
      IF (NPRINT.EQ.0) PRINT 60                                         NT 240
      DO 46 I=1,NTEQ                                                    NT 241
      IROW1=NTEQA(I)                                                    NT 242
      VLT=RHNT(I)*SI(IROW1)*WLAM                                        NT 243
      CUX=EINC(IROW1)*WLAM                                              NT 244
      YMIT=CUX/VLT                                                      NT 245
      ZPED=VLT/CUX                                                      NT 246
      IROW2=ITAG(IROW1)                                                 NT 247
      PWR=.5*REAL(VLT*CONJG(CUX))                                       NT 248
      PNLS=PNLS-PWR                                                     NT 249
   46 IF (NPRINT.EQ.0) PRINT 62, IROW2,IROW1,VLT,CUX,ZPED,YMIT,PWR      NT 250
      IF (NTSC.EQ.0) GO TO 49                                           NT 251
      DO 47 I=1,NTSC                                                    NT 252
      IROW1=NTSCA(I)                                                    NT 253
      VLT=VSRC(I)                                                       NT 254
      CUX=EINC(IROW1)*WLAM                                              NT 255
      YMIT=CUX/VLT                                                      NT 256
```

```
          ZPED=VLT/CUX                                                  NT 257
          IROW2=ITAG(IROW1)                                             NT 258
          PWR=.5*REAL(VLT*CONJG(CUX))                                   NT 259
          PNLS=PNLS-PWR                                                 NT 260
   47     IF (NPRINT.EQ.0) PRINT 62, IROW2,IROW1,VLT,CUX,ZPED,YMIT,PWR  NT 261
          GO TO 49                                                      NT 262
C                                                                       NT 263
C         SOLVE FOR CURRENTS WHEN NO NETWORKS ARE PRESENT               NT 264
C                                                                       NT 265
   48     CALL SOLGF (CM,CMB,CMC,CMD,EINC,IP,NP,N1,N,MP,M1,M,NEQ,NEQ2)  NT 266
          CALL CABC (EINC)                                              NT 267
          NTSC=0                                                        NT 268
   49     IF (NSANT+NVQD.EQ.0) RETURN                                   NT 269
          PRINT 63                                                      NT 270
          PRINT 60                                                      NT 271
          IF (NSANT.EQ.0) GO TO 56                                      NT 272
          DO 55 I=1,NSANT                                               NT 273
          ISC1=ISANT(I)                                                 NT 274
          VLT=VSANT(I)                                                  NT 275
          IF (NTSC.EQ.0) GO TO 51                                       NT 276
          DO 50 J=1,NTSC                                                NT 277
          IF (NTSCA(J).EQ.ISC1) GO TO 52                                NT 278
   50     CONTINUE                                                      NT 279
   51     CUX=EINC(ISC1)*WLAM                                           NT 280
          IROW1=0                                                       NT 281
          GO TO 54                                                      NT 282
   52     IROW1=NDIMNP-J                                                NT 283
          CUX=RHNX(IROW1)                                               NT 284
          DO 53 J=1,NTEQ                                                NT 285
   53     CUX=CUX-CMN(J,IROW1)*RHNT(J)                                  NT 286
          CUX=(EINC(ISC1)+CUX)*WLAM                                     NT 287
   54     YMIT=CUX/VLT                                                  NT 288
          ZPED=VLT/CUX                                                  NT 289
          PWR=.5*REAL(VLT*CONJG(CUX))                                   NT 290
          PIN=PIN+PWR                                                   NT 291
          IF (IROW1.NE.0) PNLS=PNLS+PWR                                 NT 292
          IROW2=ITAG(ISC1)                                              NT 293
   55     PRINT 62, IROW2,ISC1,VLT,CUX,ZPED,YMIT,PWR                    NT 294
   56     IF (NVQD.EQ.0) RETURN                                         NT 295
          DO 57 I=1,NVQD                                                NT 296
          ISC1=IVQD(I)                                                  NT 297
          VLT=VQD(I)                                                    NT 298
          CUX=CMPLX(AIR(ISC1),AII(ISC1))                                NT 299
          YMIT=CMPLX(BIR(ISC1),BII(ISC1))                               NT 300
          ZPED=CMPLX(CIR(ISC1),CII(ISC1))                               NT 301
          PWR=SI(ISC1)*TP*.5                                            NT 302
          CUX=(CUX-YMIT*SIN(PWR)+ZPED*COS(PWR))*WLAM                    NT 303
          YMIT=CUX/VLT                                                  NT 304
          ZPED=VLT/CUX                                                  NT 305
          PWR=.5*REAL(VLT*CONJG(CUX))                                   NT 306
          PIN=PIN+PWR                                                   NT 307
          IROW2=ITAG(ISC1)                                              NT 308
   57     PRINT 64, IROW2,ISC1,VLT,CUX,ZPED,YMIT,PWR                    NT 309
          RETURN                                                        NT 310
C                                                                       NT 311
   58     FORMAT (///,3X,47HMAXIMUM RELATIVE ASYMMETRY OF THE DRIVING POINT, NT 312
         121H ADMITTANCE MATRIX IS,E10.3,13H FOR SEGMENTS,I5,4H AND,I5,/,3X, NT 313
         225HRMS RELATIVE ASYMMETRY IS,E10.3)                           NT 314
   59     FORMAT (1X,44HERROR - - NETWORK ARRAY DIMENSIONS TOO SMALL)   NT 315
   60     FORMAT (/,3X,3HTAG,3X,4HSEG.,4X,15HVOLTAGE (VOLTS),9X,14HCURRENT ( NT 316
         1AMPS),9X,16HIMPEDANCE (OHMS),8X,17HADMITTANCE (MHOS),6X,5HPOWER,/, NT 317
         23X,3HNO.,3X,3HNO.,4X,4HREAL,8X,5HIMAG.,3(7X,4HREAL,8X,5HIMAG.),5X, NT 318
         37H(WATTS))                                                    NT 319
   61     FORMAT (///,27X,66H- - - STRUCTURE EXCITATION DATA AT NETWORK CONN NT 320
```

```
     1ECTION POINTS - - -)                                             NT 321
  62  FORMAT (2(1X,I5),9E12.5)                                         NT 322
  63  FORMAT (///,42X,36H- - - ANTENNA INPUT PARAMETERS - - -)         NT 323
  64  FORMAT (1X,I5,2H *,I4,9E12.5)                                    NT 324
      END                                                              NT 325-
```

NFPAT

PURPOSE

To compute and print the near E or H field over a range of points.

METHOD

The range of points in rectangular or spherical coordinates is obtained from parameters in COMMON/FPAT/. Subroutine NEFLD is called for near E field and NHFLD is called for near H field.

SYMBOL DICTIONARY

CPH	= cos ϕ
CTH	= cos θ
DXNR	= increment for x in rectangular coordinates or R in spherical coordinates
DYNR	= increment for y in rectangular coordinates or ϕ in spherical coordinates
DZNR	= increment for z in rectangular coordinates or θ in spherical coordinates
EX, EY, EZ	= x, y and z components of E or H
NEAR	= 0 for rectangular coordinates 1 for spherical coordinates
NFEH	= 0 for near E field 1 for near H field
NRX, NRY, NRZ	= number of values for x, y and z or R, ϕ, θ
SPH	= sin ϕ
STH	= sin θ
TA	= $\pi/180$
XNR	= initial x or R
XNRT	= x or R
XOB	= x
YNR	= initial y or ϕ
YNRT	= y or ϕ
YOB	= y
ZNR	= initial z or θ
ZNRT	= z or θ
ZOB	= z

```
      SUBROUTINE NFPAT                                                  NP   1
C     COMPUTE NEAR E OR H FIELDS OVER A RANGE OF POINTS                 NP   2
      COMPLEX EX,EY,EZ                                                  NP   3
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 NP   4
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( NP   5
     2300),WLAM,IPSYM                                                   NP   6
      COMMON /FPAT/ NTH,NPH,IPD,IAVP,INOR,IAX,THETS,PHIS,DTH,DPH,RFLD,GN NP   7
     1OR,CLT,CHT,EPSR2,SIG2,IXTYP,XPR6,PINR,PNLR,PLOSS,NEAR,NFEH,NRX,NRY NP   8
     2,NRZ,XNR,YNR,ZNR,DXNR,DYNR,DZNR                                   NP   9
      DATA TA/1.745329252E-02/                                          NP  10
      IF (NFEH.EQ.1) GO TO 1                                            NP  11
      PRINT 10                                                          NP  12
      GO TO 2                                                           NP  13
1     PRINT 12                                                          NP  14
2     ZNRT=ZNR-DZNR                                                     NP  15
      DO 9 I=1,NRZ                                                      NP  16
      ZNRT=ZNRT+DZNR                                                    NP  17
      IF (NEAR.EQ.0) GO TO 3                                            NP  18
      CTH=COS(TA*ZNRT)                                                  NP  19
      STH=SIN(TA*ZNRT)                                                  NP  20
3     YNRT=YNR-DYNR                                                     NP  21
      DO 9 J=1,NRY                                                      NP  22
      YNRT=YNRT+DYNR                                                    NP  23
      IF (NEAR.EQ.0) GO TO 4                                            NP  24
      CPH=COS(TA*YNRT)                                                  NP  25
      SPH=SIN(TA*YNRT)                                                  NP  26
4     XNRT=XNR-DXNR                                                     NP  27
      DO 9 KK=1,NRX                                                     NP  28
      XNRT=XNRT+DXNR                                                    NP  29
      IF (NEAR.EQ.0) GO TO 5                                            NP  30
      XOB=XNRT*STH*CPH                                                  NP  31
      YOB=XNRT*STH*SPH                                                  NP  32
      ZOB=XNRT*CTH                                                      NP  33
      GO TO 6                                                           NP  34
5     XOB=XNRT                                                          NP  35
      YOB=YNRT                                                          NP  36
      ZOB=ZNRT                                                          NP  37
6     TMP1=XOB/WLAM                                                     NP  38
      TMP2=YOB/WLAM                                                     NP  39
      TMP3=ZOB/WLAM                                                     NP  40
      IF (NFEH.EQ.1) GO TO 7                                            NP  41
      CALL NEFLD (TMP1,TMP2,TMP3,EX,EY,EZ)                              NP  42
      GO TO 8                                                           NP  43
7     CALL NHFLD (TMP1,TMP2,TMP3,EX,EY,EZ)                              NP  44
8     TMP1=CABS(EX)                                                     NP  45
      TMP2=CANG(EX)                                                     NP  46
      TMP3=CABS(EY)                                                     NP  47
      TMP4=CANG(EY)                                                     NP  48
      TMP5=CABS(EZ)                                                     NP  49
      TMP6=CANG(EZ)                                                     NP  50
      PRINT 11, XOB,YOB,ZOB,TMP1,TMP2,TMP3,TMP4,TMP5,TMP6               NP  51
9     CONTINUE                                                          NP  52
      RETURN                                                            NP  53
C                                                                       NP  54
10    FORMAT (///,35X,32H- - - NEAR ELECTRIC FIELDS - - -,//,12X,14H-  L NP  55
     1OCATION  -,21X,8H-  EX  -,15X,8H-  EY  -,15X,8H-  EZ  -,/,8X,1HX,1 NP  56
     20X,1HY,10X,1HZ,10X,9HMAGNITUDE,3X,5HPHASE,6X,9HMAGNITUDE,3X,5HPHAS NP  57
     3E,6X,9HMAGNITUDE,3X,5HPHASE,/,6X,6HMETERS,5X,6HMETERS,5X,6HMETERS, NP  58
     48X,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3 NP  59
     5X,7HDEGREES)                                                      NP  60
11    FORMAT (2X,3(2X,F9.4),1X,3(3X,E11.4,2X,F7.2))                     NP  61
12    FORMAT (///,35X,32H- - - NEAR MAGNETIC FIELDS - - -,//,12X,14H-  L NP  62
     1OCATION  -,21X,8H-  HX  -,15X,8H-  HY  -,15X,8H-  HZ  -,/,8X,1HX,1 NP  63
     20X,1HY,10X,1HZ,10X,9HMAGNITUDE,3X,5HPHASE,6X,9HMAGNITUDE,3X,5HPHAS NP  64
```

```
     3E,6X,9HMAGNITUDE,3X,5HPHASE,/,6X,6HMETERS,5X,6HMETERS,5X,6HMETERS,  NP  65
     49X,6HAMPS/M,3X,7HDEGREES,7X,6HAMPS/M,3X,7HDEGREES,7X,6HAMPS/M,3X,7  NP  66
     5HDEGREES)                                                           NP  67
      END                                                                 NP  68-
```

NHFLD

PURPOSE

To compute the near magnetic field due to currents induced on a structure.

CODING

NH28 - NH56 Near H field due to currents on segments is computed.

NH29 - NH40 Each segment is checked to determine whether the field observation point (XOB, YOB, ZOB) falls within the segment volume. If it does, AX is set to the radius of that segment. AX is then sent to routine HSFLD as the radius of the observation segment to avoid a singularity in the field.

NH41 - NH56 Loop computing the field contribution of each segment.

NH42 - NH49 Parameters of source segment are stored in COMMON/DATAJ/.

NH50 HSFLD stores the magnetic field due to constant, sin ks, and cos ks currents in COMMON/DATAJ/.

NH54 - NH56 The field components are multiplied by the coefficients of the constant, sin ks, and cos ks components of the total segment current, and the field is summed.

NH58 - NH78 Near H fields due to patch currents are computed.

NH62 - NH71 Parameters of source patch are set in COMMON/DATAJ/.

NH72 H field is computed by HINTG.

NH76 - NH78 H fields due to $\hat{t}_1$ and $\hat{t}_2$ current components are multiplied by the current strengths and summed.

SYMBOL DICTIONARY

ACX = constant component of the segment current at NH51; $\hat{t}_1$ component of patch current at NH74

AX = segment radius when the field evaluation point falls within a segment volume

BCX = sin ks component of segment current at NH52; $\hat{t}_2$ component of patch current at NH75

CCX = cos ks component of segment current at NH53

HX, HY, HZ = total H field

T1X
T1Y
T1Z } = arrays for $\hat{t}_1$

T1XJ
T1YJ
T1ZJ } = $\hat{t}_1$ for patch I

T2X
T2Y
T2Z } = arrays for $\hat{t}_2$

T2XJ
T2YJ
T2ZJ } = $\hat{t}_2$ for patch I

XOB
YOB
ZOB } = field evaluation point

ZP = coordinates of the field evaluation point, z or ρ^2, in a cylindrical coordinate system centered on the source segment.

CONSTANTS

0.5001 = fraction of segment length used to test whether the field evaluation point falls within a segment

0.9 = fraction of segment radius used to test whether the field evaluation point falls within a segment

```
      SUBROUTINE NHFLD (XOB,YOB,ZOB,HX,HY,HZ)
C
C     NHFLD COMPUTES THE NEAR FIELD AT SPECIFIED POINTS IN SPACE AFTER
C     THE STRUCTURE CURRENTS HAVE BEEN COMPUTED.
C
      COMPLEX HX,HY,HZ,CUR,ACX,BCX,CCX,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,E
     1ZC
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      COMMON /CRNT/ AIR(300),AII(300),BIR(300),BII(300),CIR(300),CII(300
     1),CUR(900)
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      DIMENSION CAB(1), SAB(1)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), XS(1), Y
     1S(1), ZS(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (XS,X), (YS,Y), (ZS,Z)
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y
     1J,IND1), (T2ZJ,IND2)
      EQUIVALENCE (CAB,ALP), (SAB,BET)
      HX=(0.,0.)
      HY=(0.,0.)
      HZ=(0.,0.)
      AX=0.
      IF (N.EQ.0) GO TO 4
      DO 1 I=1,N
      XJ=XOB-X(I)
      YJ=YOB-Y(I)
      ZJ=ZOB-Z(I)
      ZP=CAB(I)*XJ+SAB(I)*YJ+SALP(I)*ZJ
      IF (ABS(ZP).GT.0.5001*SI(I)) GO TO 1
      ZP=XJ*XJ+YJ*YJ+ZJ*ZJ-ZP*ZP
      XJ=BI(I)
      IF (ZP.GT.0.9*XJ*XJ) GO TO 1
      AX=XJ
      GO TO 2
1     CONTINUE
2     DO 3 I=1,N
      S=SI(I)
      B=BI(I)
      XJ=X(I)
      YJ=Y(I)
      ZJ=Z(I)
      CABJ=CAB(I)
      SABJ=SAB(I)
      SALPJ=SALP(I)
      CALL HSFLD (XOB,YOB,ZOB,AX)
      ACX=CMPLX(AIR(I),AII(I))
      BCX=CMPLX(BIR(I),BII(I))
      CCX=CMPLX(CIR(I),CII(I))
      HX=HX+EXK*ACX+EXS*BCX+EXC*CCX
      HY=HY+EYK*ACX+EYS*BCX+EYC*CCX
3     HZ=HZ+EZK*ACX+EZS*BCX+EZC*CCX
      IF (M.EQ.0) RETURN
4     JC=N
      JL=LD+1
      DO 5 I=1,M
      JL=JL-1
      S=BI(JL)
      XJ=X(JL)
      YJ=Y(JL)
```

```
      ZJ=Z(JL)                                                          NH  65
      T1XJ=T1X(JL)                                                      NH  66
      T1YJ=T1Y(JL)                                                      NH  67
      T1ZJ=T1Z(JL)                                                      NH  68
      T2XJ=T2X(JL)                                                      NH  69
      T2YJ=T2Y(JL)                                                      NH  70
      T2ZJ=T2Z(JL)                                                      NH  71
      CALL HINTG (XOB,YOB,ZOB)                                          NH  72
      JC=JC+3                                                           NH  73
      ACX=T1XJ*CUR(JC-2)+T1YJ*CUR(JC-1)+T1ZJ*CUR(JC)                    NH  74
      BCX=T2XJ*CUR(JC-2)+T2YJ*CUR(JC-1)+T2ZJ*CUR(JC)                    NH  75
      HX=HX+ACX*EXK+BCX*EXS                                             NH  76
      HY=HY+ACX*EYK+BCX*EYS                                             NH  77
5     HZ=HZ+ACX*EZK+BCX*EZS                                             NH  78
      RETURN                                                            NH  79
      END                                                               NH  80-
```

PATCH (entry SUBPH)

PURPOSE

To generate patch data for surfaces.

METHOD

The code from PA14 to PA129 generates data for a single new patch or multiple patches. There are four options for defining a single patch, as illustrated in Figure 5 of Part III. For a single patch, NX is zero and NY is NS + 1 where NS is the parameter from the SP input card and is shown on Figure 5. Rectangular, triangular or quadrilateral patches are defined by the coordinates of three or four corners in the parameters X1 though Z4. In the arbitrary shape option (Figure 5A in Part III) the center of the patch is X1, Y1, Z1; α is X2; β is Y2; and the area is Z2. The patch data is stored in COMMON/DATA/ from the top of the arrays downward (see Section III).

The code from PA131 to PA190 divides a patch into four patches and is used when a wire connects to a patch. If NY is equal to zero the patch NX is divided into four patches that become patches NX through NX + 3. Patches following NX are shifted in the arrays in COMMON/DATA/ to leave space for the three additional patches. If NY is greater than zero, patch NX is left in the arrays but four new patches to replace it are added to the end of the arrays. The z coordinate of patch NX is then changed to 10,000 at PA189.

SYMBOL DICTIONARY

MI	= array index for patch data
MIA	= array index for patch data
NTP	= patch type (NY for a single patch)
NX	= zero for a single patch. For multiple patches NX is defined in Figure 6 of Part III. After ENTRY SUBPH, NX is the number of the patch to be divided
S1X, S1Y, S1Z	= vector from corner 1 to corner 2
S2X, S2Y, S2Z	= vector from corner 2 to corner 3
SALN	= ± 1 from array SALP
SALPN	= factor in computing center of mass of quadrilateral

PATCH

XA	$= \lvert\bar{S}_1 \times \bar{S}_2\rvert$ = area of rectangle or twice area of triangle (PA53)
XN2, YN2, ZN2	$= \bar{S}_3 \times \bar{S}_4$ at PA79 to PA81. Line PA89 checks that the four corners are coplanar by the test $(\bar{S}_1 \times \bar{S}_2).(\bar{S}_3 \times \bar{S}_4)/\lvert\bar{S}_1 \times \bar{S}_2\rvert\lvert\bar{S}_3 \times \bar{S}_4\rvert > 0.9998$
XNV, YNV, ZNV	= unit vector normal to the patch at PA54 to PA56
XS, YS, ZS	= patch center at PA151 to PA153
XST	$= \lvert\bar{S}_1 \times \bar{S}_2\rvert$ at PA57

CONSTANTS

$0.9998 \approx \cos(1.°)$ in test for planar patch

```
      SUBROUTINE PATCH (NX,NY,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4)      PA   1
C     PATCH GENERATES AND MODIFIES PATCH GEOMETRY DATA                  PA   2
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 PA   3
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( PA   4
     2300),WLAM,IPSYM                                                   PA   5
      COMMON /ANGL/ SALP(300)                                           PA   6
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)          PA   7
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON PA   8
     12), (T2Z,ITAG)                                                    PA   9
C     NEW PATCHES.  FOR NX=0, NY=1,2,3,4 PATCH IS (RESPECTIVELY)        PA  10
C     ARBITRARY, RECTAGULAR, TRIANGULAR, OR QUADRILATERAL.              PA  11
C     FOR NX AND NY .GT. 0 A RECTANGULAR SURFACE IS PRODUCED WITH       PA  12
C     NX BY NY RECTANGULAR PATCHES.                                     PA  13
      M=M+1                                                             PA  14
      MI=LD+1-M                                                         PA  15
      NTP=NY                                                            PA  16
      IF (NX.GT.0) NTP=2                                                PA  17
      IF (NTP.GT.1) GO TO 2                                             PA  18
      X(MI)=X1                                                          PA  19
      Y(MI)=Y1                                                          PA  20
      Z(MI)=Z1                                                          PA  21
      BI(MI)=Z2                                                         PA  22
      ZNV=COS(X2)                                                       PA  23
      XNV=ZNV*COS(Y2)                                                   PA  24
      YNV=ZNV*SIN(Y2)                                                   PA  25
      ZNV=SIN(X2)                                                       PA  26
      XA=SQRT(XNV*XNV+YNV*YNV)                                          PA  27
      IF (XA.LT.1.E-6) GO TO 1                                          PA  28
      T1X(MI)=-YNV/XA                                                   PA  29
      T1Y(MI)=XNV/XA                                                    PA  30
      T1Z(MI)=0.                                                        PA  31
      GO TO 6                                                           PA  32
1     T1X(MI)=1.                                                        PA  33
      T1Y(MI)=0.                                                        PA  34
      T1Z(MI)=0.                                                        PA  35
      GO TO 6                                                           PA  36
2     S1X=X2-X1                                                         PA  37
      S1Y=Y2-Y1                                                         PA  38
      S1Z=Z2-Z1                                                         PA  39
      S2X=X3-X2                                                         PA  40
      S2Y=Y3-Y2                                                         PA  41
      S2Z=Z3-Z2                                                         PA  42
      IF (NX.EQ.0) GO TO 3                                              PA  43
      S1X=S1X/NX                                                        PA  44
      S1Y=S1Y/NX                                                        PA  45
      S1Z=S1Z/NX                                                        PA  46
      S2X=S2X/NY                                                        PA  47
      S2Y=S2Y/NY                                                        PA  48
      S2Z=S2Z/NY                                                        PA  49
3     XNV=S1Y*S2Z-S1Z*S2Y                                               PA  50
      YNV=S1Z*S2X-S1X*S2Z                                               PA  51
      ZNV=S1X*S2Y-S1Y*S2X                                               PA  52
      XA=SQRT(XNV*XNV+YNV*YNV+ZNV*ZNV)                                  PA  53
      XNV=XNV/XA                                                        PA  54
      YNV=YNV/XA                                                        PA  55
      ZNV=ZNV/XA                                                        PA  56
      XST=SQRT(S1X*S1X+S1Y*S1Y+S1Z*S1Z)                                 PA  57
      T1X(MI)=S1X/XST                                                   PA  58
      T1Y(MI)=S1Y/XST                                                   PA  59
      T1Z(MI)=S1Z/XST                                                   PA  60
      IF (NTP.GT.2) GO TO 4                                             PA  61
      X(MI)=X1+.5*(S1X+S2X)                                             PA  62
      Y(MI)=Y1+.5*(S1Y+S2Y)                                             PA  63
      Z(MI)=Z1+.5*(S1Z+S2Z)                                             PA  64
```

```
      BI(MI)=XA
      GO TO 6
    4 IF (NTP.EQ.4) GO TO 5
      X(MI)=(X1+X2+X3)/3.
      Y(MI)=(Y1+Y2+Y3)/3.
      Z(MI)=(Z1+Z2+Z3)/3.
      BI(MI)=.5*XA
      GO TO 6
    5 S1X=X3-X1
      S1Y=Y3-Y1
      S1Z=Z3-Z1
      S2X=X4-X1
      S2Y=Y4-Y1
      S2Z=Z4-Z1
      XN2=S1Y*S2Z-S1Z*S2Y
      YN2=S1Z*S2X-S1X*S2Z
      ZN2=S1X*S2Y-S1Y*S2X
      XST=SQRT(XN2*XN2+YN2*YN2+ZN2*ZN2)
      SALPN=1./(3.*(XA+XST))
      X(MI)=(XA*(X1+X2+X3)+XST*(X1+X3+X4))*SALPN
      Y(MI)=(XA*(Y1+Y2+Y3)+XST*(Y1+Y3+Y4))*SALPN
      Z(MI)=(XA*(Z1+Z2+Z3)+XST*(Z1+Z3+Z4))*SALPN
      BI(MI)=.5*(XA+XST)
      S1X=(XNV*XN2+YNV*YN2+ZNV*ZN2)/XST
      IF (S1X.GT.0.9998) GO TO 6
      PRINT 14
      STOP
    6 T2X(MI)=YNV*T1Z(MI)-ZNV*T1Y(MI)
      T2Y(MI)=ZNV*T1X(MI)-XNV*T1Z(MI)
      T2Z(MI)=XNV*T1Y(MI)-YNV*T1X(MI)
      SALP(MI)=1.
      IF (NX.EQ.0) GO TO 8
      M=M+NX*NY-1
      XN2=X(MI)-S1X-S2X
      YN2=Y(MI)-S1Y-S2Y
      ZN2=Z(MI)-S1Z-S2Z
      XS=T1X(MI)
      YS=T1Y(MI)
      ZS=T1Z(MI)
      XT=T2X(MI)
      YT=T2Y(MI)
      ZT=T2Z(MI)
      MI=MI+1
      DO 7 IY=1,NY
      XN2=XN2+S2X
      YN2=YN2+S2Y
      ZN2=ZN2+S2Z
      DO 7 IX=1,NX
      XST=IX
      MI=MI-1
      X(MI)=XN2+XST*S1X
      Y(MI)=YN2+XST*S1Y
      Z(MI)=ZN2+XST*S1Z
      BI(MI)=XA
      SALP(MI)=1
      T1X(MI)=XS
      T1Y(MI)=YS
      T1Z(MI)=ZS
      T2X(MI)=XT
      T2Y(MI)=YT
    7 T2Z(MI)=ZT
    8 IPSYM=0
      [illegible]
      [illegible]
```

```
      RETURN                                                            PA 129
C     DIVIDE PATCH FOR WIRE CONNECTION                                  PA 130
      ENTRY SUBPH(NX,NY,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4)            PA 131
      IF (NY.GT.0) GO TO 10                                             PA 132
      IF (NX.EQ.M) GO TO 10                                             PA 133
      NXP=NX+1                                                          PA 134
      IX=LD-M                                                           PA 135
      DO 9 IY=NXP,M                                                     PA 136
      IX=IX+1                                                           PA 137
      NYP=IX-3                                                          PA 138
      X(NYP)=X(IX)                                                      PA 139
      Y(NYP)=Y(IX)                                                      PA 140
      Z(NYP)=Z(IX)                                                      PA 141
      BI(NYP)=BI(IX)                                                    PA 142
      SALP(NYP)=SALP(IX)                                                PA 143
      T1X(NYP)=T1X(IX)                                                  PA 144
      T1Y(NYP)=T1Y(IX)                                                  PA 145
      T1Z(NYP)=T1Z(IX)                                                  PA 146
      T2X(NYP)=T2X(IX)                                                  PA 147
      T2Y(NYP)=T2Y(IX)                                                  PA 148
9     T2Z(NYP)=T2Z(IX)                                                  PA 149
10    MI=LD+1-NX                                                        PA 150
      XS=X(MI)                                                          PA 151
      YS=Y(MI)                                                          PA 152
      ZS=Z(MI)                                                          PA 153
      XA=BI(MI)*.25                                                     PA 154
      XST=SQRT(XA)*.5                                                   PA 155
      S1X=T1X(MI)                                                       PA 156
      S1Y=T1Y(MI)                                                       PA 157
      S1Z=T1Z(MI)                                                       PA 158
      S2X=T2X(MI)                                                       PA 159
      S2Y=T2Y(MI)                                                       PA 160
      S2Z=T2Z(MI)                                                       PA 161
      SALN=SALP(MI)                                                     PA 162
      XT=XST                                                            PA 163
      YT=XST                                                            PA 164
      IF (NY.GT.0) GO TO 11                                             PA 165
      MIA=MI                                                            PA 166
      GO TO 12                                                          PA 167
11    M=M+1                                                             PA 168
      MP=MP+1                                                           PA 169
      MIA=LD+1-M                                                        PA 170
12    DO 13 IX=1,4                                                      PA 171
      X(MIA)=XS+XT*S1X+YT*S2X                                           PA 172
      Y(MIA)=YS+XT*S1Y+YT*S2Y                                           PA 173
      Z(MIA)=ZS+XT*S1Z+YT*S2Z                                           PA 174
      BI(MIA)=XA                                                        PA 175
      T1X(MIA)=S1X                                                      PA 176
      T1Y(MIA)=S1Y                                                      PA 177
      T1Z(MIA)=S1Z                                                      PA 178
      T2X(MIA)=S2X                                                      PA 179
      T2Y(MIA)=S2Y                                                      PA 180
      T2Z(MIA)=S2Z                                                      PA 181
      SALP(MIA)=SALN                                                    PA 182
      IF (IX.EQ.2) YT=-YT                                               PA 183
      IF (IX.EQ.1.OR.IX.EQ.3) XT=-XT                                    PA 184
      MIA=MIA-1                                                         PA 185
13    CONTINUE                                                          PA 186
      M=M+3                                                             PA 187
      IF (NX.LE.MP) MP=MP+3                                             PA 188
      IF (NY.GT.0) Z(MI)=10000.                                         PA 189
      RETURN                                                            PA 190
C                                                                       PA 191
14    FORMAT (62H ERROR -- CORNERS OF QUADRILATERAL PATCH DO NOT LIE IN PA 192
```

```
193        1A PLANE)                                                    PA 193
194         END                                                         PA 194-
```

PCINT

PURPOSE

To compute the interaction matrix elements representing the electric field, tangent to a segment connected to a surface, due to the current on the four patches around the connection point.

METHOD

The four patches at the base of a connected wire are located as shown in figure 10 with respect to the vectors $\hat{t}_1$ and $\hat{t}_2$, where patch numbers indicate the order of the patches in the data arrays. The position of a point on the surface is defined by $\overline{\rho}\,(S_1,S_2) = \overline{\rho}_0 + S_1\hat{t}_1 + S_2\hat{t}_2$, where $\overline{\rho}_0$ is the position of the center of the four patches where the wire connects, and S_1 and S_2 are coordinates measured from the center. The current over the surface is represented by $\overline{J}(S_1,S_2)$, the currents at the centers of the four patches are

$$\overline{J}_1 = \overline{J}(d,d)$$
$$\overline{J}_2 = \overline{J}(-d,d)$$
$$\overline{J}_3 = \overline{J}(-d,-d)$$
$$\overline{J}_4 = \overline{J}(d,-d)$$

and the current at the base of the segment, flowing onto the surface, is I_0. The current interpolation function is then

$$\overline{J}(S_1,S_2) = \left[\overline{f}(S_1,S_2) - \sum_{i=1}^{4} g_i(S_1,S_2)\,\overline{f}_i\right] I_0 + \sum_{i=1}^{4} g_i(S_1,S_2)\overline{J}_i\ ,$$

where

$$\overline{f}(S_1,S_2) = \frac{S_1\hat{t}_1 + S_2\hat{t}_2}{2\pi(S_1^{\ 2} + S_2^{\ 2})}$$

$$\overline{f}_1 = \overline{f}(d,d) = (\hat{t}_1 + \hat{t}_2)/(4\pi d)$$
$$\overline{f}_2 = \overline{f}(-d,d) = (-\hat{t}_1 + \hat{t}_2)/(4\pi d)$$
$$\overline{f}_3 = \overline{f}(-d,-d) = (-\hat{t}_1 - \hat{t}_2)/(4\pi d)$$
$$\overline{f}_4 = \overline{f}(d,-d) = (\hat{t}_1 - \hat{t}_2)/(4\pi d)$$

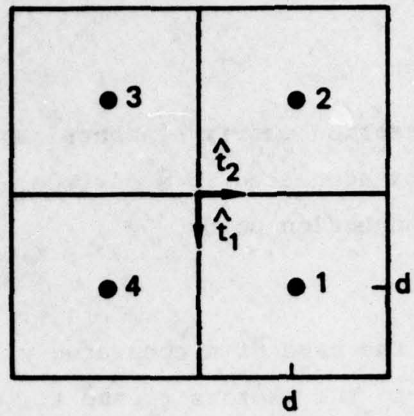


Figure 10. Patches at a Wire Connection Point.

$$g_1(S_1,S_2) = (d + S_1)(d + S_2)/(4d^2)$$
$$g_2(S_1,S_2) = (d - S_1)(d + S_2)/(4d^2)$$
$$g_3(S_1,S_2) = (d - S_1)(d - S_2)/(4d^2)$$
$$g_4(S_1,S_2) = (d + S_1)(d - S_2)/(4d^2)$$

If $\overline{\Gamma}_1(\overline{\rho})dA$ and $\overline{\Gamma}_2(\overline{\rho})dA$ are the electric fields at the center of the connected segment due to unit currents at $\overline{\rho}$ on the surface dA, flowing in the directions $\hat{t}_1$ and $\hat{t}_2$, respectively, the nine matrix elements to be computed are

$$E_1 = \int_S g_1(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_1(\overline{\rho})dA$$

$$E_2 = \int_S g_2(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_1(\overline{\rho})dA$$

$$E_3 = \int_S g_3(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_1(\overline{\rho})dA$$

$$E_4 = \int_S g_4(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_1(\overline{\rho})dA$$

$$E_5 = \int_S g_1(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_2(\overline{\rho})dA$$

$$E_6 = \int_S g_2(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_2(\overline{\rho})dA$$

$$E_7 = \int_S g_3(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_2(\overline{\rho})dA$$

$$E_8 = \int_S g_4(S_1,S_2)\ \hat{i} \cdot \overline{\Gamma}_2(\overline{\rho})dA$$

$$E_9 = \int_S \left\{ \left[\overline{h}(S_1,S_2) \cdot \hat{t}_1\right] \left[\hat{i} \cdot \overline{\Gamma}_1(\overline{\rho})\right] + \left[\overline{h}(S_1,S_2) \cdot \hat{t}_2\right] \left[\hat{i} \cdot \overline{\Gamma}_2(\overline{\rho})\right] \right\} dA$$

where

$$\overline{h}(S_1,S_2) = \overline{f}(S_1,S_2) - \sum_{i=1}^{4} g_i(S_1,S_2)\overline{f}_i ,$$

and where $\hat{i}$ = the unit vector in the direction of the connected segment.

The integration is over the total area of the four patches and is performed by numerical quadrature. The number of increments in S_1 and S_2 used in integration is set by the variable NINT. When PCINT is called, the parameters in COMMON/DATAJ/ have the values for the first connected patch. During integration, these parameters are set for each integration patch. At the end of PCINT, they are reset to their original values.

SYMBOL DICTIONARY

CABI	= x component of $\hat{i}$
D	= d
DA	= area of the surface element used in integration
DS	= width of the surface element of area DA
E	= array used to return the values E_1, E_2, ..., E_9
EXK, EYK, EZK	= x, y, and z components of $\overline{\Gamma}_1(\overline{\rho})$DA at PC50; at PC51, EXK is set to $\hat{i} \cdot \overline{\Gamma}_1(\overline{\rho})$DA
EXS, EYS, EZS	= x, y, and z components of $\overline{\Gamma}_2(\overline{\rho})$DA at PC50; at PC52, EXS is set to $\hat{i} \cdot \overline{\Gamma}_2(\overline{\rho})$DA

E1 = E_1
E2 = E_2
E3 = E_3
E4 = E_4
E5 = E_5
E6 = E_6
E7 = E_7
E8 = E_8
E9 = E_9
FCON = $1/(4\pi d)$ factor in $\overline{f}_1$, $\overline{f}_2$, ...
F1 = $\overline{h}(S_1,S_2) \cdot \hat{t}_1$
F2 = $\overline{h}(S_1,S_2) \cdot \hat{t}_2$
GCON = $1/(4d^2)$ factor in $g_1(S_1,S_2)$, ...
G1 = $g_1(S_1,S_2)$
G2 = $g_2(S_1,S_2)$
G3 = $g_3(S_1,S_2)$
G4 = $g_4(S_1,S_2)$
I1 = DO loop index
I2 = DO loop index
NINT = number of steps in S_1 and S_3 used in approximating the integrals for E_1, E_2, ..., E_9
S = area of each of the four patches at PC11; area of the surface element used in integration at PC20
SABI = y component of $\hat{i}$
SALPI = z component of $\hat{i}$
S1 = S_1
S2 = S_2
S2X = initial value of S_2
TPI = 2π
T1XJ, T1YJ, T1ZJ = x, y, and z components of $\hat{t}_1$
T2XJ, T2YJ, T2ZJ = x, y, and z components of $\hat{t}_2$
XI = x coordinate of the center of the connected segment

XJ, YJ, ZJ	= center of first patch above PC41; center of integration element below PC41
XS	= x component of $\overline{\rho}(S_1,S_2)$
XSS	= initial x coordinate of $\overline{\rho}(S_1,S_2)$
XXJ, XYJ, XZJ	= initial value of XJ, YJ, ZJ saved
X1	= x component of $\overline{\rho}(d,d)$ used as reference for computing $\overline{\rho}(S_1,S_2)$
YI	= y coordinate of the center of the connected segment
YS	= y component of $\overline{\rho}(S_1,S_2)$
YSS	= initial y component of $\overline{\rho}(S_1,S_2)$
Y1	= y component of $\overline{\rho}(d,d)$
ZI	= z coordinate of the center of the connected segment
ZS	= z component of $\overline{\rho}(S_1,S_2)$
ZSS	= initial z component of $\overline{\rho}(S_1,S_2)$
Z1	= z component of $\overline{\rho}(d,d)$

```
      SUBROUTINE PCINT (XI,YI,ZI,CABI,SABI,SALPI,E)
C     INTEGRATE OVER PATCHES AT WIRE CONNECTION POINT
      COMPLEX EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC,E,E1,E2,E3,E4,E5,E6,E7
     1,E8,E9
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      DIMENSION E(9)
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y
     1J,IND1), (T2ZJ,IND2)
      DATA TPI/6.283185308/,NINT/10/
      D=SQRT(S)*.5
      DS=4.*D/FLOAT(NINT)
      DA=DS*DS
      GCON=1./S
      FCON=1./(2.*TPI*D)
      XXJ=XJ
      XYJ=YJ
      XZJ=ZJ
      XS=S
      S=DA
      S1=D+DS*.5
      XSS=XJ+S1*(T1XJ+T2XJ)
      YSS=YJ+S1*(T1YJ+T2YJ)
      ZSS=ZJ+S1*(T1ZJ+T2ZJ)
      S1=S1+D
      S2X=S1
      E1=(0.,0.)
      E2=(0.,0.)
      E3=(0.,0.)
      E4=(0.,0.)
      E5=(0.,0.)
      E6=(0.,0.)
      E7=(0.,0.)
      E8=(0.,0.)
      E9=(0.,0.)
      DO 1 I1=1,NINT
      S1=S1-DS
      S2=S2X
      XSS=XSS-DS*T1XJ
      YSS=YSS-DS*T1YJ
      ZSS=ZSS-DS*T1ZJ
      XJ=XSS
      YJ=YSS
      ZJ=ZSS
      DO 1 I2=1,NINT
      S2=S2-DS
      XJ=XJ-DS*T2XJ
      YJ=YJ-DS*T2YJ
      ZJ=ZJ-DS*T2ZJ
      CALL UNERE (XI,YI,ZI)
      EXK=EXK*CABI+EYK*SABI+EZK*SALPI
      EXS=EXS*CABI+EYS*SABI+EZS*SALPI
      G1=(D+S1)*(D+S2)*GCON
      G2=(D-S1)*(D+S2)*GCON
      G3=(D-S1)*(D-S2)*GCON
      G4=(D+S1)*(D-S2)*GCON
      F2=(S1*S1+S2*S2)*TPI
      F1=S1/F2-(G1-G2-G3+G4)*FCON
      F2=S2/F2-(G1+G2-G3-G4)*FCON
      E1=E1+EXK*G1
      E2=E2+EXK*G2
      E3=E3+EXK*G3
      E4=E4+EXK*G4
      E5=E5+EXS*G1
```

```
      E6=E6+EXS*G2                                                      PC  65
      E7=E7+EXS*G3                                                      PC  66
      E8=E8+EXS*G4                                                      PC  67
    1 E9=E9+EXK*F1+EXS*F2                                               PC  68
      E(1)=E1                                                           PC  69
      E(2)=E2                                                           PC  70
      E(3)=E3                                                           PC  71
      E(4)=E4                                                           PC  72
      E(5)=E5                                                           PC  73
      E(6)=E6                                                           PC  74
      E(7)=E7                                                           PC  75
      E(8)=E8                                                           PC  76
      E(9)=E9                                                           PC  77
      XJ=XXJ                                                            PC  78
      YJ=XYJ                                                            PC  79
      ZJ=XZJ                                                            PC  80
      S=XS                                                              PC  81
      RETURN                                                            PC  82
      END                                                               PC  83-
```

PRNT

PURPOSE

To set up the formats for printing a record of three integers, six floating point numbers, and a Hollerith string, where the variables equal to zero are replaced by blanks. This routine is used by LOAD in printing the impedance data table.

METHOD

A variable format is used to generate the record with arbitrary blank fill. Elements of the format are picked from the array IFORM in the DATA statement. Through IF statements operating on the subroutine input quantities, this routine chooses the desired format elements and builds the format in the array IVAR. The program is divided into two sections: the first builds the integer part of the format and the second the floating point part.

SYMBOL DICTIONARY

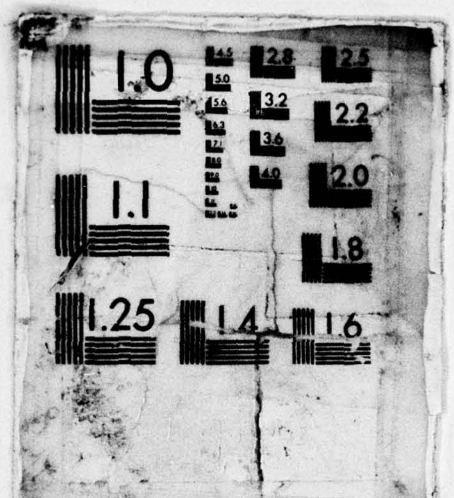
ABS	= external routine (absolute value)
FL	= elements of this array are set equal to the floating point input quantities FL1 - FL6
FLT	= array of non-zero floating point input quantities to be printed
FL1, FL2, FL3, FL4, FL5, FL6	= input floating point quantities
HALL	= 4H ALL (Hollerith ALL)
I	= DO loop index
IA	= input Hollerith string (array)
ICHAR	= number of characters in the input Hollerith string
IFORM	= array containing format elements
IN	= array set equal to input integer quantities (IN1 - IN3)
INT	= non-zero integer quantities to be printed
IN1, IN2, IN3	= input integer quantities
IVAR	= variable format array

I1 = DO loop limit
J = implied DO loop index
K = index parameter
L = implied DO loop index
NCPW = number of Hollerith characters per computer word
NFLT = floating point print index, number of non-zero reals
NINT = integer print index; number of non-zero integers
NWORDS = number of computer words in the input Hollerith string

```
      SUBROUTINE PRNT (IN1,IN2,IN3,FL1,FL2,FL3,FL4,FL5,FL6,IA,ICHAR)
C
C     PRNT SETS UP THE PRINT FORMATS FOR IMPEDANCE LOADING
C
      DIMENSION IVAR(13), IA(1), IFORM(8), IN(3), INT(3), FL(6), FLT(6)
      INTEGER HALL
      DATA IFORM/5H(/3X,,3HI5,,3H5X,,3HA5,,6HE13.4,,4H13X,,3H3X,,5H2A10)
     1/
C
C     NUMBER OF CHARACTERS PER COMPUTER WORD IS NCPW
C
      DATA NCPW/10/,HALL/4H ALL/
      NWORDS=(ICHAR-1)/NCPW+1
      IN(1)=IN1
      IN(2)=IN2
      IN(3)=IN3
      FL(1)=FL1
      FL(2)=FL2
      FL(3)=FL3
      FL(4)=FL4
      FL(5)=FL5
      FL(6)=FL6
C
C     INTEGER FORMAT
C
      NINT=0
      IVAR(1)=IFORM(1)
      K=1
      I1=1
      IF (.NOT.(IN1.EQ.0.AND.IN2.EQ.0.AND.IN3.EQ.0)) GO TO 1
      INT(1)=HALL
      NINT=1
      I1=2
      K=K+1
      IVAR(K)=IFORM(4)
1     DO 3 I=I1,3
      K=K+1
      IF (IN(I).EQ.0) GO TO 2
      NINT=NINT+1
      INT(NINT)=IN(I)
      IVAR(K)=IFORM(2)
      GO TO 3
2     IVAR(K)=IFORM(3)
3     CONTINUE
      K=K+1
      IVAR(K)=IFORM(7)
C
C     FLOATING POINT FORMAT
C
      NFLT=0
      DO 5 I=1,6
      K=K+1
      IF (ABS(FL(I)).LT.1.E-20) GO TO 4
      NFLT=NFLT+1
      FLT(NFLT)=FL(I)
      IVAR(K)=IFORM(5)
      GO TO 5
4     IVAR(K)=IFORM(6)
5     CONTINUE
      K=K+1
      IVAR(K)=IFORM(7)
      K=K+1
      IVAR(K)=IFORM(8)
      PRINT IVAR, (INT(I),I=1,NINT),(FLT(J),J=1,NFLT),(IA(L),L=1,NWORDS)
```

```
65      RETURN                                                    PR  65
66      END                                                       PR  66-
```

AD-A075 289 NUMERICAL ELECTROMAGNETIC CODE (NEC)-METHOD OF MOMENTS 5/6
A USER-ORIENTED CO..(U) NAVAL OCEAN SYSTEMS CENTER SAN
DIEGO CA G J BURKE ET AL. 18 JUL 77 NOSC/TD-116-VOL-1
UNCLASSIFIED AFWL-TR-76-320-VOL-1 F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART

QDSRC

PURPOSE

To fill the excitation array for a current slope discontinuity voltage source.

METHOD

The current slope discontinuity voltage source is described in section IV-1 of Part I.

CODING

QD22 - QD25 The connection number for end 1 of segment IS is temporarily set to 0, and TBF is called to generate the function $f^*_\ell(s)$ for ℓ - IS. The zero in the second argument of TBF causes f^*_ℓ to go to zero at the first end of segment IS rather than the usual non-zero value that allows for current flowing onto the wire end cap.

QD26 - QD31 β_ℓ is computed and other quantities set.

QD32 - QD119 This loop computes the fields due to each segment on which f^*_ℓ is non-zero.

QD33 - QD77 Parameters of the source segment are stored in COMMON/ DATAJ/. Flags for the extended thin wire approximation are set as in routine CMSET.

QD78 - QD91 This loop evaluates the electric field on each segment.

QD95 - QD116 This loop evaluates the magnetic field at each patch.

SYMBOL DICTIONARY

AI = radius of segment on which field is evaluated.

CABI = x component of unit vector in the direction of segment I

CCJ = CCJX = -j/60

CURD = β_ℓ

E = array of segment and patch excitation fields

ETC, ETK, ETS = E field tangent to a segment or H field components on a patch due to cosine, constant, and sine current components, respectively, on a segment

I1 = array index for patch excitation

IJ = flag which, if zero, indicates that the field is being evaluated on the source segment

IPR = temporary storage of connection number

IS = segment which has the source location on end 1

J = source segment number

SABI = y component of unit vector in the direction of segment I

T1X, T1Y, T1Z, T2X, T2Y, T2Z = arrays of components of $\hat{t}_1$ and $\hat{t}_2$ for patches

TP = 2π

TX, TY, TZ = components of $\hat{t}_1$ or $\hat{t}_2$ for patches

V = source voltage

XI, YI, ZI = coordinates of point where field is evaluated; XI is also used in the test for the extended thin wire approximation for the electric field

CONSTANTS

0.01666666667 = 1/60

0.999999 = minimum XI for the extended thin wire approximation (maximum angle = 0.08 degrees)

6.283185308 = 2π

```
      SUBROUTINE QDSRC (IS,V,E)                                         QD   1
C     FILL INCIDENT FIELD ARRAY FOR CHARGE DISCONTINUITY VOLTAGE SOURCE QD   2
      COMPLEX VQDS,CURD,CCJ,V,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC,ETK,ET QD   3
     1S,ETC,VSANT,VQD,E,ZARRAY                                          QD   4
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 QD   5
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( QD   6
     2300),WLAM,IPSYM                                                   QD   7
      COMMON /VSORC/ VQD(30),VSANT(30),VQDS(30),IVQD(30),ISANT(30),IQDS( QD   8
     130),NVQD,NSANT,NQDS                                               QD   9
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP QD  10
     1CON(10),NPCON                                                     QD  11
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ QD  12
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND                            QD  13
      COMMON /ANGL/ SALP(300)                                           QD  14
      COMMON /ZLOAD/ ZARRAY(300),NLOAD,NLODF                            QD  15
      DIMENSION CCJX(2), E(1), CAB(1), SAB(1)                           QD  16
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1)          QD  17
      EQUIVALENCE (CCJ,CCJX), (CAB,ALP), (SAB,BET)                      QD  18
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON QD  19
     12), (T2Z,ITAG)                                                    QD  20
      DATA TP/6.283185308/,CCJX/0.,-.01666666667/                       QD  21
      I=ICON1(IS)                                                       QD  22
      ICON1(IS)=0                                                       QD  23
      CALL TBF (IS,0)                                                   QD  24
      ICON1(IS)=I                                                       QD  25
      S=SI(IS)*.5                                                       QD  26
      CURD=CCJ*V/((ALOG(2.*S/BI(IS))-1.)*(BX(JSNO)*COS(TP*S)+CX(JSNO)*SI QD  27
     1N(TP*S))*WLAM)                                                    QD  28
      NQDS=NQDS+1                                                       QD  29
      VQDS(NQDS)=V                                                      QD  30
      IQDS(NQDS)=IS                                                     QD  31
      DO 20 JX=1,JSNO                                                   QD  32
      J=JCO(JX)                                                         QD  33
      S=SI(J)                                                           QD  34
      B=BI(J)                                                           QD  35
      XJ=X(J)                                                           QD  36
      YJ=Y(J)                                                           QD  37
      ZJ=Z(J)                                                           QD  38
      CABJ=CAB(J)                                                       QD  39
      SABJ=SAB(J)                                                       QD  40
      SALPJ=SALP(J)                                                     QD  41
      IF (IEXK.EQ.0) GO TO 16                                           QD  42
      IPR=ICON1(J)                                                      QD  43
      IF (IPR) 1,6,2                                                    QD  44
1     IPR=-IPR                                                          QD  45
      IF (-ICON1(IPR).NE.J) GO TO 7                                     QD  46
      GO TO 4                                                           QD  47
2     IF (IPR.NE.J) GO TO 3                                             QD  48
      IF (CABJ*CABJ+SABJ*SABJ.GT.1.E-8) GO TO 7                         QD  49
      GO TO 5                                                           QD  50
3     IF (ICON2(IPR).NE.J) GO TO 7                                      QD  51
4     XI=ABS(CABJ*CAB(IPR)+SABJ*SAB(IPR)+SALPJ*SALP(IPR))               QD  52
      IF (XI.LT.0.999999) GO TO 7                                       QD  53
      IF (ABS(BI(IPR)/B-1.).GT.1.E-6) GO TO 7                           QD  54
5     IND1=0                                                            QD  55
      GO TO 8                                                           QD  56
6     IND1=1                                                            QD  57
      GO TO 8                                                           QD  58
7     IND1=2                                                            QD  59
8     IPR=ICON2(J)                                                      QD  60
      IF (IPR) 9,14,10                                                  QD  61
9     IPR=-IPR                                                          QD  62
      IF (-ICON2(IPR).NE.J) GO TO 15                                    QD  63
      GO TO 12                                                          QD  64
```

```
   10 IF (IPR.NE.J) GO TO 11
      IF (CABJ*CABJ+SABJ*SABJ.GT.1.E-8) GO TO 15
      GO TO 13
   11 IF (ICON1(IPR).NE.J) GO TO 15
   12 XI=ABS(CABJ*CAB(IPR)+SABJ*SAB(IPR)+SALPJ*SALP(IPR))
      IF (XI.LT.0.999999) GO TO 15
      IF (ABS(BI(IPR)/B-1.).GT.1.E-6) GO TO 15
   13 IND2=0
      GO TO 16
   14 IND2=1
      GO TO 16
   15 IND2=2
   16 CONTINUE
      DO 17 I=1,N
      IJ=I-J
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)
      AI=BI(I)
      CALL EFLD (XI,YI,ZI,AI,IJ)
      CABI=CAB(I)
      SABI=SAB(I)
      SALPI=SALP(I)
      ETK=EXK*CABI+EYK*SABI+EZK*SALPI
      ETS=EXS*CABI+EYS*SABI+EZS*SALPI
      ETC=EXC*CABI+EYC*SABI+EZC*SALPI
   17 E(I)=E(I)-(ETK*AX(JX)+ETS*BX(JX)+ETC*CX(JX))*CURD
      IF (M.EQ.0) GO TO 19
      IJ=LD+1
      I1=N
      DO 18 I=1,M
      IJ=IJ-1
      XI=X(IJ)
      YI=Y(IJ)
      ZI=Z(IJ)
      CALL HSFLD (XI,YI,ZI,0.)
      I1=I1+1
      TX=T2X(IJ)
      TY=T2Y(IJ)
      TZ=T2Z(IJ)
      ETK=EXK*TX+EYK*TY+EZK*TZ
      ETS=EXS*TX+EYS*TY+EZS*TZ
      ETC=EXC*TX+EYC*TY+EZC*TZ
      E(I1)=E(I1)+(ETK*AX(JX)+ETS*BX(JX)+ETC*CX(JX))*CURD*SALP(IJ)
      I1=I1+1
      TX=T1X(IJ)
      TY=T1Y(IJ)
      TZ=T1Z(IJ)
      ETK=EXK*TX+EYK*TY+EZK*TZ
      ETS=EXS*TX+EYS*TY+EZS*TZ
      ETC=EXC*TX+EYC*TY+EZC*TZ
   18 E(I1)=E(I1)+(ETK*AX(JX)+ETS*BX(JX)+ETC*CX(JX))*CURD*SALP(IJ)
   19 IF (NLOAD.GT.0.OR.NLODF.GT.0) E(J)=E(J)+ZARRAY(J)*CURD*(AX(JX)+CX(
     1JX))
   20 CONTINUE
      RETURN
      END
```

RDPAT

PURPOSE

To compute and print radiated field quantities.

METHOD

The quantities computed and the output formats depend on the options selected by the first integer (IFAR) and fourth integer (IPD, IAVP, INOR, IAX) on the RP card (see Part III). These quantities are defined as follows:

(1) Power Gain

In the direction (θ,ϕ)

$$G_p(\theta,\phi) = 4\pi \frac{P_\Omega(\theta,\phi)}{P_{in}},$$

where $P_\Omega(\theta,\phi)$ is the power radiated per unit solid angle in the given direction, and P_{in} is the total power accepted by the antenna. Therefore, $P_{in} = (1/2)\text{Re}(VI^*)$, where V is the applied source voltage, and

$$P_\Omega(\theta,\phi) = (1/2)\ R^2\text{Re}(\bar{E} \times \bar{H}^*) = \frac{R^2}{2\eta}\bar{E} \cdot \bar{E}^*,$$

where R is the observation sphere radius. Since the electric field calculated by FFLD (call it $\bar{E}'$) does not include $\exp(-jkR)/(R/\lambda)$,

$$\bar{E} = \frac{\exp(-jkR)}{R/\lambda}\bar{E}'$$

and

$$P_\Omega = \frac{\lambda^2}{2\eta}(\bar{E}' \cdot \bar{E}'^*) \quad .$$

Thus,

$$G_p(\theta,\phi) = \frac{2\pi\lambda^2}{\eta P_{in}} (\bar{E}' \cdot \bar{E}'^*)$$

in terms of the program variables.

(2) Directive Gain

In the direction (θ,ϕ),

$$G_d(\theta,\phi) = 4\pi \frac{P_\Omega(\theta,\phi)}{P_{rad}}$$

where P_{rad} is the total power radiated by the antenna. The only difference from power gain is that P_{in} is replaced by P_{rad}, and $P_{rad} = P_{in} - P_{loss}$, where P_{loss} is calculated as the power lost in distributed and lumped loads on the structure and in the networks loads.

(3) Component Gain

The gains are also calculated for separate, orthogonal field components (u, v). In this case, $\bar{E}' \cdot \bar{E}'^*$ is replaced by $E'_u E'^*_u$ or $E'_v E'^*_v$, and the total gain is the sum of the two components.

(4) Average Gain

The user specifies a range and number of points in theta and phi that in turn specify the total solid angle covered, Ω, and the sampling density for the integral in the expression for average gain:

$$G_{av} = \frac{\int_\Omega G_p \, d\Omega}{\Omega}$$

The trapezoidal rule is used in evaluating the integral.

(5) Normalized Gain

Normalized gain is simply the gain divided by its maximum value or some value specified by the user.

The discussion of gains applies only to the case of a structure used as a radiating antenna. For the case of an incident plane wave, the program constants are defined such that the value of σ/λ^2 is printed under the heading "GAIN." The calculation is

$$\frac{\sigma}{\lambda^2} = \frac{4\pi R^2}{\lambda^2} \frac{W_{scat}}{W_{inc}} = \frac{4\pi}{\bar{E}_{inc} \cdot \bar{E}^*_{inc}} (\bar{E}'_{scat} \cdot \bar{E}'^*_{scat}) ,$$

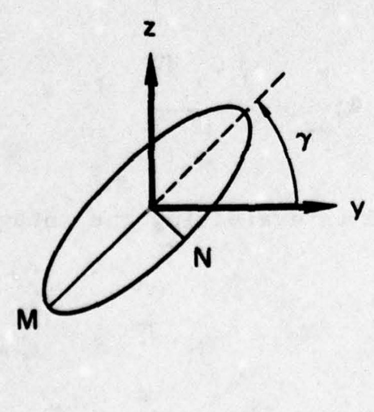
where W_{scat} is the scattered power per unit area at distance R in a given direction, W_{inc} is the power per unit area of the incident plane wave, and the primes on the electric fields specify the fields used in the program as defined above. For the case of a Hertzian dipole used as a source, the gain equations are used; however, P_{in} is equal to the total power radiated by the Hertzian source. That is

$$P_{in} = \frac{\pi\eta}{3} \left| \frac{I\ell}{\lambda} \right|^2 ,$$

where the quantity $I\ell$ is an input quantity.

(6) Elliptic Polarization

Elliptic polarization parameters are calculated as follows:



$$M = [(E_{ym} \cos\gamma + E_{zm} \cos\xi \sin\gamma)^2 + E_{zm}^2 \sin^2\xi \sin^2\gamma]^{1/2} ,$$

$$N = [E_{ym} \sin\gamma - E_{zm} \cos\xi \cos\gamma)^2 + E_{zm}^2 \sin^2\xi \cos^2\gamma]^{1/2} ,$$

where

$$E_y = E_{ym} \exp[j(\omega t - kx)] ,$$

$$E_z = E_{zm} \exp[j(\omega t - kx + \xi)] ,$$

and γ is given by

$$\tan 2y = \frac{2E_{ym}E_{zm}\cos\xi}{E_{ym}^2 - E_{zm}^2}$$

In this routine, the coordinates y and z above are replaced by θ and ϕ, respectively.

The field is computed by FFLD at RD74 for space wave or by GFLD at RD76 for space and ground wave. Elliptic polarization parameters are computed from RD87 to RD118. RD127 to RD137 stores gain in the array GAIN for normalization. The integral of radiated power for the average gain calculation is summed at RD140 to RD147. Fields and gain are printed at RD162 for space wave or RD165 for ground wave. Average gain is computed and printed from RD168 to RD173. Normalized gain is printed from RD174 to RD208.

SYMBOL DICTIONARY

AXRAT	= N/M (elliptic axial ratio)
CHT	= height of cliff in meters
CLT	= distance in meters of cliff edge from origin
DA	= element of solid angle for average gain summation
DFAZ	= phase difference between E_θ and E_ϕ for elliptic polarization

DPH	= increment for ϕ
DTH	= increment for θ
EMAJR2	= M^2 (M = major axis)
EMINR2	= N^2
EPH	= E_ϕ (phi component of electric field, with or without the term $\exp(-jkR)/(R/\lambda)$ depending on return from GFLD or FFLD)
EPHA	= phase angle of EPH
EPHM	= $\lvert EPH \rvert$
EPHM2	= $\lvert EPH \rvert^2$
EPSR	= relative dielectric constant
EPSR2	= relative dielectric constant of second medium
ERD	= radial electric field for ground wave
ERDA	= phase of ERD
ERDM	= $\lvert ERD \rvert$
ETH	= E_θ
ETHA	= phase of E_θ
ETHM	= $\lvert E_\theta \rvert$
ETHM2	= $\lvert E_\theta \rvert^2$
EXRA	= phase of $\exp(-jkR)$
EXRM	= $1/R$
GCON	= factor multiplying $\lvert E \rvert^2$ to yield gain or σ/λ^2
GCOP	= GCON except when GCON yields directive gain; then GCOP remains power gain
GMAX	= value used for normalized gain
GNH	= horizontal gain in decibels, ϕ component
GNMJ	= major axis gain in decibels
GNMN	= minor axis gain in decibels
GNOR	= if non-zero, equals input gain quantity
GNV	= vertical gain (θ)
GTOT	= total gain
IAVP	= flag for average gain
IAX	= flag for gain type
IFAR	= first integer from RP card

INOR	=	integer to select normalized gain
IPD	=	flag to select power or directive gain
IXTYP	=	excitation type
NORMAX	=	dimension of FNORM (maximum number of gain values that will be stored for normalization)
NPH	=	number of ϕ values
NTH	=	number of θ values
PHA	=	ϕ in radians
PHI	=	ϕ in degrees
PHIS	=	initial ϕ
PI	=	π
PINR	=	input power for current element source
PINT	=	summation variable for average gain
PLOSS	=	power dissipated in structure loads
PNLR	=	power dissipated in networks and transmission lines
PRAD	=	power radiated by the antenna
RFLD	=	if non-zero, equal to the observation distance in meters
SIG	=	conductivity of ground (mhos/m)
SIG2	=	conductivity of second medium (mhos/m)
STILTA	=	sin γ; γ is tilt angle of the polarization ellipse
TA	=	$\pi/180$
TD	=	$180/\pi$
THA	=	θ in radians
THET	=	θ in degrees
THETS	=	initial θ
TILTA	=	γ (tilt angle of ellipse)
XPR6	=	minor axis of polarization ellipse or strength of current element source

CONSTANTS

1.745329252E-2	=	$\pi/180$
1.E-20	=	small value test

RDPAT

1.E-5	= small value test
-1.E10	= near minus infinity
3.141592654	= π
376.73	= $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$
394.51	= $\pi\eta_0/3$
57.2957795	= $180/\pi$
59.96	= $\eta_0/(2\pi)$
90.01	= test value for angle exceeding 90 degrees

```
      SUBROUTINE RDPAT                                                  RD   1
C     COMPUTE RADIATION PATTERN, GAIN, NORMALIZED GAIN                  RD   2
      INTEGER HPOL,HBLK,HCIR,HCLIF                                      RD   3
      COMPLEX ETH,EPH,ERD,ZRATI,ZRATI2,T1,FRATI                         RD   4
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 RD   5
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( RD   6
     2300),WLAM,IPSYM                                                   RD   7
      COMMON /SAVE/ IP(600),KCOM,COM(13,5),EPSR,SIG,SCRWLT,SCRWRT,FMHZ  RD   8
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR, RD   9
     1IPERF,T1,T2                                                       RD  10
      COMMON /FPAT/ NTH,NPH,IPD,IAVP,INOR,IAX,THETS,PHIS,DTH,DPH,RFLD,GN RD  11
     1OR,CLT,CHT,EPSR2,SIG2,IXTYP,XPR6,PINR,PNLR,PLOSS,NEAR,NFEH,NRX,NRY RD  12
     2,NRZ,XNR,YNR,ZNR,DXNR,DYNR,DZNR                                   RD  13
      COMMON /SCRATM/ GAIN(1200)                                        RD  14
      DIMENSION IGTP(4), IGAX(4), IGNTP(10), HPOL(3)                    RD  15
      DATA HPOL/6HLINEAR,5HRIGHT,4HLEFT/,HBLK,HCIR/1H ,6HCIRCLE/        RD  16
      DATA IGTP/6H    - ,6HPOWER ,6H- DIRE,6HCTIVE /                    RD  17
      DATA IGAX/6H MAJOR,6H MINOR,6H VERT.,6H HOR. /                    RD  18
      DATA IGNTP/6H MAJOR,6H AXIS ,6H MINOR,6H AXIS ,6H   VER,6HTICAL ,6 RD  19
     1H HORIZ,6HONTAL ,6H      ,6HTOTAL /                               RD  20
      DATA PI,TA,TD/3.141592654,1.745329252E-02,57.29577951/            RD  21
      DATA NORMAX/1200/                                                 RD  22
      IF (IFAR.LT.2) GO TO 2                                            RD  23
      PRINT 35                                                          RD  24
      IF (IFAR.LE.3) GO TO 1                                            RD  25
      PRINT 36, NRADL,SCRWLT,SCRWRT                                     RD  26
      IF (IFAR.EQ.4) GO TO 2                                            RD  27
1     IF (IFAR.EQ.2.OR.IFAR.EQ.5) HCLIF=HPOL(1)                         RD  28
      IF (IFAR.EQ.3.OR.IFAR.EQ.6) HCLIF=HCIR                            RD  29
      CL=CLT/WLAM                                                       RD  30
      CH=CHT/WLAM                                                       RD  31
      ZRATI2=CSQRT(1./CMPLX(EPSR2,-SIG2*WLAM*59.96))                    RD  32
      PRINT 37, HCLIF,CLT,CHT,EPSR2,SIG2                                RD  33
2     IF (IFAR.NE.1) GO TO 3                                            RD  34
      PRINT 41                                                          RD  35
      GO TO 5                                                           RD  36
3     I=2*IPD+1                                                         RD  37
      J=I+1                                                             RD  38
      ITMP1=2*IAX+1                                                     RD  39
      ITMP2=ITMP1+1                                                     RD  40
      PRINT 38                                                          RD  41
      IF (RFLD.LT.1.E-20) GO TO 4                                       RD  42
      EXRM=1./RFLD                                                      RD  43
      EXRA=RFLD/WLAM                                                    RD  44
      EXRA=-360.*(EXRA-AINT(EXRA))                                      RD  45
      PRINT 39, RFLD,EXRM,EXRA                                          RD  46
4     PRINT 40, IGTP(I),IGTP(J),IGAX(ITMP1),IGAX(ITMP2)                 RD  47
5     IF (IXTYP.EQ.0.OR.IXTYP.EQ.5) GO TO 7                             RD  48
      IF (IXTYP.EQ.4) GO TO 6                                           RD  49
      PRAD=0.                                                           RD  50
      GCON=4.*PI/(1.+XPR6*XPR6)                                         RD  51
      GCOP=GCON                                                         RD  52
      GO TO 8                                                           RD  53
6     PINR=394.51*XPR6*XPR6*WLAM*WLAM                                   RD  54
7     GCOP=WLAM*WLAM*2.*PI/(376.73*PINR)                                RD  55
      PRAD=PINR-PLOSS-PNLR                                              RD  56
      GCON=GCOP                                                         RD  57
      IF (IPD.NE.0) GCON=GCON*PINR/PRAD                                 RD  58
8     I=0                                                               RD  59
      GMAX=-1.E10                                                       RD  60
      PINT=0.                                                           RD  61
      TMP1=DPH*TA                                                       RD  62
      TMP2=.5*DTH*TA                                                    RD  63
      PHI=PHIS-DPH                                                      RD  64
```

```
      DO 29 KPH=1,NPH
      PHI=PHI+DPH
      PHA=PHI*TA
      THET=THETS-DTH
      DO 29 KTH=1,NTH
      THET=THET+DTH
      IF (KSYMP.EQ.2.AND.THET.GT.90.01.AND.IFAR.NE.1) GO TO 29
      THA=THET*TA
      IF (IFAR.EQ.1) GO TO 9
      CALL FFLD (THA,PHA,ETH,EPH)
      GO TO 10
9     CALL GFLD (RFLD/WLAM,PHA,THET/WLAM,ETH,EPH,ERD,ZRATI,KSYMP)
      ERDM=CABS(ERD)
      ERDA=CANG(ERD)
10    ETHM2=REAL(ETH*CONJG(ETH))
      ETHM=SQRT(ETHM2)
      ETHA=CANG(ETH)
      EPHM2=REAL(EPH*CONJG(EPH))
      EPHM=SQRT(EPHM2)
      EPHA=CANG(EPH)
      IF (IFAR.EQ.1) GO TO 28
C     ELLIPTICAL POLARIZATION CALC.
      IF (ETHM2.GT.1.E-20.OR.EPHM2.GT.1.E-20) GO TO 11
      TILTA=0.
      EMAJR2=0.
      EMINR2=0.
      AXRAT=0.
      ISENS=HBLK
      GO TO 16
11    DFAZ=EPHA-ETHA
      IF (EPHA.LT.0.) GO TO 12
      DFAZ2=DFAZ-360.
      GO TO 13
12    DFAZ2=DFAZ+360.
13    IF (ABS(DFAZ).GT.ABS(DFAZ2)) DFAZ=DFAZ2
      CDFAZ=COS(DFAZ*TA)
      TSTOR1=ETHM2-EPHM2
      TSTOR2=2.*EPHM*ETHM*CDFAZ
      TILTA=.5*ATGN2(TSTOR2,TSTOR1)
      STILTA=SIN(TILTA)
      TSTOR1=TSTOR1*STILTA*STILTA
      TSTOR2=TSTOR2*STILTA*COS(TILTA)
      EMAJR2=-TSTOR1+TSTOR2+ETHM2
      EMINR2=TSTOR1-TSTOR2+EPHM2
      IF (EMINR2.LT.0.) EMINR2=0.
      AXRAT=SQRT(EMINR2/EMAJR2)
      TILTA=TILTA*TD
      IF (AXRAT.GT.1.E-5) GO TO 14
      ISENS=HPOL(1)
      GO TO 16
14    IF (DFAZ.GT.0.) GO TO 15
      ISENS=HPOL(2)
      GO TO 16
15    ISENS=HPOL(3)
16    GNMJ=DB10(GCON*EMAJR2)
      GNMN=DB10(GCON*EMINR2)
      GNV=DB10(GCON*ETHM2)
      GNH=DB10(GCON*EPHM2)
      GTOT=DB10(GCON*(ETHM2+EPHM2))
      IF (INOR.LT.1) GO TO 23
      I=I+1
      IF (I.GT.NORMAX) GO TO 23
      GO TO (17,18,19,20,21), INOR
17    TSTOR1=GNMJ
```

```
      GO TO 22
   18 TSTOR1=GNMN
      GO TO 22
   19 TSTOR1=GNV
      GO TO 22
   20 TSTOR1=GNH
      GO TO 22
   21 TSTOR1=GTOT
   22 GAIN(I)=TSTOR1
      IF (TSTOR1.GT.GMAX) GMAX=TSTOR1
   23 IF (IAVP.EQ.0) GO TO 24
      TSTOR1=GCOP*(ETHM2+EPHM2)
      TMP3=THA-TMP2
      TMP4=THA+TMP2
      IF (KTH.EQ.1) TMP3=THA
      IF (KTH.EQ.NTH) TMP4=THA
      DA=ABS(TMP1*(COS(TMP3)-COS(TMP4)))
      IF (KPH.EQ.1.OR.KPH.EQ.NPH) DA=.5*DA
      PINT=PINT+TSTOR1*DA
      IF (IAVP.EQ.2) GO TO 29
   24 IF (IAX.EQ.1) GO TO 25
      TMP5=GNMJ
      TMP6=GNMN
      GO TO 26
   25 TMP5=GNV
      TMP6=GNH
   26 ETHM=ETHM*WLAM
      EPHM=EPHM*WLAM
      IF (RFLD.LT.1.E-20) GO TO 27
      ETHM=ETHM*EXRM
      ETHA=ETHA+EXRA
      EPHM=EPHM*EXRM
      EPHA=EPHA+EXRA
   27 PRINT 42, THET,PHI,TMP5,TMP6,GTOT,AXRAT,TILTA,ISENS,ETHM,ETHA,EPHM
     1,EPHA
      GO TO 29
   28 PRINT 43, RFLD,PHI,THET,ETHM,ETHA,EPHM,EPHA,ERDM,ERDA
   29 CONTINUE
      IF (IAVP.EQ.0) GO TO 30
      TMP3=THETS*TA
      TMP4=TMP3+DTH*TA*FLOAT(NTH-1)
      TMP3=ABS(DPH*TA*FLOAT(NPH-1)*(COS(TMP3)-COS(TMP4)))
      PINT=PINT/TMP3
      TMP3=TMP3/PI
      PRINT 44, PINT,TMP3
   30 IF (INOR.EQ.0) GO TO 34
      IF (ABS(GNOR).GT.1.E-20) GMAX=GNOR
      ITMP1=(INOR-1)*2+1
      ITMP2=ITMP1+1
      PRINT 45, IGNTP(ITMP1),IGNTP(ITMP2),GMAX
      ITMP2=NPH*NTH
      IF (ITMP2.GT.NORMAX) ITMP2=NORMAX
      ITMP1=(ITMP2+2)/3
      ITMP2=ITMP1*3-ITMP2
      ITMP3=ITMP1
      ITMP4=2*ITMP1
      IF (ITMP2.EQ.2) ITMP4=ITMP4-1
      DO 31 I=1,ITMP1
      ITMP3=ITMP3+1
      ITMP4=ITMP4+1
      J=(I-1)/NTH
      TMP1=THETS+FLOAT(I-J*NTH-1)*DTH
      TMP2=PHIS+FLOAT(J)*DPH
      J=(ITMP3-1)/NTH
```

```
      TMP3=THETS+FLOAT(ITMP3-J*NTH-1)*DTH
      TMP4=PHIS+FLOAT(J)*DPH
      J=(ITMP4-1)/NTH
      TMP5=THETS+FLOAT(ITMP4-J*NTH-1)*DTH
      TMP6=PHIS+FLOAT(J)*DPH
      TSTOR1=GAIN(I)-GMAX
      IF (I.EQ.ITMP1.AND.ITMP2.NE.0) GO TO 32
      TSTOR2=GAIN(ITMP3)-GMAX
      PINT=GAIN(ITMP4)-GMAX
   31 PRINT 46, TMP1,TMP2,TSTOR1,TMP3,TMP4,TSTOR2,TMP5,TMP6,PINT
      GO TO 34
   32 IF (ITMP2.EQ.2) GO TO 33
      TSTOR2=GAIN(ITMP3)-GMAX
      PRINT 46, TMP1,TMP2,TSTOR1,TMP3,TMP4,TSTOR2
      GO TO 34
   33 PRINT 46, TMP1,TMP2,TSTOR1
   34 RETURN
C
   35 FORMAT (///,31X,39H- - - FAR FIELD GROUND PARAMETERS - - -,//)
   36 FORMAT (40X,25HRADIAL WIRE GROUND SCREEN,/,40X,I5,6H WIRES,/,40X,1
     12HWIRE LENGTH=,F8.2,7H METERS,/,40X,12HWIRE RADIUS=,E10.3,7H METER
     2S)
   37 FORMAT (40X,A6,6H CLIFF,/,40X,14HEDGE DISTANCE=,F9.2,7H METERS,/,4
     10X,7HHEIGHT=,F8.2,7H METERS,/,40X,15HSECOND MEDIUM -,/,40X,27HRELA
     2TIVE DIELECTRIC CONST.=,F7.3,/,40X,13HCONDUCTIVITY=,E10.3,5H MHOS)
   38 FORMAT (///,48X,30H- - - RADIATION PATTERNS - - -)
   39 FORMAT (54X,6HRANGE=,E13.6,7H METERS,/,54X,12HEXP(-JKR)/R=,E12.5,9
     1H AT PHASE,F7.2,8H DEGREES,/)
   40 FORMAT (/,2X,14H- - ANGLES - -,7X,2A6,7HGAINS -,7X,24H- - - POLARI
     1ZATION - - -,4X,20H- - - E(THETA) - - -,4X,16H- - - E(PHI) - -,2H
     2-,/,2X,5HTHETA,5X,3HPHI,7X,A6,2X,A6,3X,5HTOTAL,6X,5HAXIAL,5X,4HTIL
     3T,3X,5HSENSE,2(5X,9HMAGNITUDE,4X,6HPHASE ),/,2(1X,7HDEGREES,1X),3(
     46X,2HDB),8X,5HRATIO,5X,4HDEG.,8X,2(6X,7HVOLTS/M,4X,7HDEGREES))
   41 FORMAT (///,28X,40H - - - RADIATED FIELDS NEAR GROUND - - -,//,8X,
     120H- - - LOCATION - - -,10X,16H- - E(THETA) - -,8X,14H- - E(PHI) -
     2 -,8X,17H- - E(RADIAL) - -,/,7X,3HRHO,6X,3HPHI,9X,1HZ,12X,3HMAG,6X
     3,5HPHASE,9X,3HMAG,6X,5HPHASE,9X,3HMAG,6X,5HPHASE,/,5X,6HMETERS,3X,
     47HDEGREES,4X,6HMETERS,8X,7HVOLTS/M,3X,7HDEGREES,6X,7HVOLTS/M,3X,7H
     5DEGREES,6X,7HVOLTS/M,3X,7HDEGREES,/)
   42 FORMAT (1X,F7.2,F9.2,3X,3F8.2,F11.5,F9.2,2X,A6,2(E15.5,F9.2))
   43 FORMAT (3X,F9.2,2X,F7.2,2X,F9.2,1X,3(3X,E11.4,2X,F7.2))
   44 FORMAT (//,3X,19HAVERAGE POWER GAIN=,E12.5,7X, 31HSOLID ANGLE USED
     1 IN AVERAGING=(,F7.4,16H)*PI STERADIANS.,//)
   45 FORMAT (//,37X,31H- - - - NORMALIZED GAIN - - - -,//,37X,2A6,4HGAI
     1N,/,38X,22HNORMALIZATION FACTOR =,F9.2,3H DB,//,3(4X,14H- - ANGLES
     2 - -,6X,4HGAIN,7X),/,3(4X,5HTHETA,5X,3HPHI,8X,2HDB,8X),/,3(3X,7HDE
     3GREES,2X,7HDEGREES,16X))
   46 FORMAT (3(1X,2F9.2,1X,F9.2,6X))
      END
```

REBLK

PURPOSE

To read the matrix B by blocks of rows and write it by blocks of columns.

METHOD

When ICASX is 3 or 4 subroutine CMNGF writes B to file 14 by blocks of rows. Filling B by rows is convenient since the field of a single segment may contribute to several columns. However, blocks of columns are needed when $A^{-1}B$ is computed. Hence the format is converted.

NBBX is the number of block of B stored by rows and NBBL is the number of blocks stored by columns. The loop from RB16 to RB23 reads file 14 and stores the elements for block NPB of columns. This process is repeated for each of the NBBL blocks of columns.

SYMBOL DICTIONARY

B = array for blocks of columns of B

BX = array for blocks of rows of B

N2C = number of columns in B

NB = number of rows in B

NBX = number of rows in blocks of rows of B (NPBX)

NPB = number of columns in blocks of columns (NPBL or NLBL for last block)

NPX = NPBX or NLBX for last block of rows

```
      SUBROUTINE REBLK (B,BX,NB,NBX,N2C)
C     REBLOCK ARRAY B IN N.G.F. SOLUTION FROM BLOCKS OF ROWS ON TAPE14
C     TO BLOCKS OF COLUMNS ON TAPE16
      COMPLEX B,BX
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      DIMENSION B(NB,1), BX(NBX,1)
      REWIND 16
      NIB=0
      NPB=NPBL
      DO 3 IB=1,NBBL
      IF (IB.EQ.NBBL) NPB=NLBL
      REWIND 14
      NIX=0
      NPX=NPBX
      DO 2 IBX=1,NBBX
      IF (IBX.EQ.NBBX) NPX=NLBX
      READ (14) ((BX(I,J),I=1,NPX),J=1,N2C)
      DO 1 I=1,NPX
      IX=I+NIX
      DO 1 J=1,NPB
1     B(IX,J)=BX(I,J+NIB)
2     NIX=NIX+NPBX
      WRITE (16) ((B(I,J),I=1,NB),J=1,NPB)
3     NIB=NIB+NPBL
      REWIND 14
      REWIND 16
      RETURN
      END
```

REFLC

PURPOSE

To generate geometry data for structures having plane or cylindrical symmetry by forming symmetric images of a previously defined structure unit.

METHOD

The first part of the code, from statement RE20 to RE153, forms plane symmetric structures by reflecting segments and patches in the coordinate planes. The reflection planes are selected by the formal parameters IX, IY, and IZ. If IZ is greater than zero, an image of the existing segments and patches is formed by reflection in the x-y plane, which will be called reflection along the z axis. Next, if IY is greater than zero, an image of the existing segments and patches, including those generated in the previous step by reflection along the z axis, is formed by reflection along the y axis. Finally, if IX is greater than zero, an image of all segments and patches, including any previously formed by reflection along the z and y axes, is formed by reflection along the x axis. Any combination of zero and non-zero values of IX, IY, and IZ may be used to generate structures with one, two, or three planes of symmetry. Tag numbers of image segments are incremented by ITX from tags of the original segments, except that tags of zero are not incremented. After each reflection in a coordinate plane, ITX is doubled. Thus, if ITX is initially greater than the largest tag of the existing segments, no duplicate tags will be formed by reflection in one, two, or three planes.

The code from RE157 to RE204 forms cylindrically symmetric structures by forming images of previously defined segments and patches rotated about the z axis. The number of images, including the original structure, is selected by NOP in the formal parameters. The angle by which each image is rotated about the z axis from the previous image is computed as 2π/NOP, so that the images are uniformly distributed about the z axis. Tag numbers of segments are incremented by ITX, except that tags of zero are not incremented.

When REFLC is used to form structures with either plane or cylindrical symmetry, the data in COMMON/DATA/ is set so that the program will take advantage of symmetry in filling and factoring the matrix. This is done by setting N equal to the total number of segments but leaving NP equal to the number of segments in the original structure unit that was reflected or

rotated. The symmetry flag IPSYM is also set to indicate the type of symmetry: positive values indicating plane symmetry and negative values cylindrical symmetry. These symmetry conditions may later be changed if the structure is modified in such a way that symmetry is destroyed.

SYMBOL DICTIONARY

ABS = external routine (absolute value)
COS = external routine (cosine)
CS = cos (2π/NOP)
E1 = segment coordinate (temporary storage)
E2 = segment coordinate (temporary storage)
FNOP = NOP
I = DO loop index
ITAGI = segment tag (temporary storage)
ITI = segment tag increment
ITX = segment tag increment
IX = flag for reflection along x axis
IY = flag for reflection along y axis
IZ = flag for reflection along z axis
J = array location for new patch data
K = segment index and array location for old patch data
NOP = number of sections in cylindrically symmetric structure
NX = segment index and array location for new patch data
NNX = array location for old patch
SAM = 2π/NOP
SIN = external routine (sine)
SS = sin (2π/NOP)
T1X, T1Y, T1Z, T2X, T2Y, T2Z = x, y, z components of $\hat{t}_1$ and $\hat{t}_2$
XK = x coordinate of segment
X2(I) = x coordinate of end two of segment I
YK = y coordinate of segment

Y2(I) = y coordinate of end two of segment I

Z2(I) = z coordinate of end two of segment I

CONSTANTS

1.E-6 = tolerance in test for zero

1.E-5 = tolerance in test for zero

6.283185308 = 2π

```
      SUBROUTINE REFLC (IX,IY,IZ,ITX,NOP)
C
C     REFLC REFLECTS PARTIAL STRUCTURE ALONG X,Y, OR Z AXES OR ROTATES
C     STRUCTURE TO COMPLETE A SYMMETRIC STRUCTURE.
C
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /ANGL/ SALP(300)
      DIMENSION T1X(1), T1Y(1), T1Z(1), T2X(1), T2Y(1), T2Z(1), X2(1), Y
     12(1), Z2(1)
      EQUIVALENCE (T1X,SI), (T1Y,ALP), (T1Z,BET), (T2X,ICON1), (T2Y,ICON
     12), (T2Z,ITAG), (X2,SI), (Y2,ALP), (Z2,BET)
      NP=N
      MP=M
      IPSYM=0
      ITI=ITX
      IF (IX.LT.0) GO TO 19
      IF (NOP.EQ.0) RETURN
      IPSYM=1
      IF (IZ.EQ.0) GO TO 6
C
C     REFLECT ALONG Z AXIS
C
      IPSYM=2
      IF (N.LT.N2) GO TO 3
      DO 2 I=N2,N
      NX=I+N-N1
      E1=Z(I)
      E2=Z2(I)
      IF (ABS(E1)+ABS(E2).GT.1.E-5.AND.E1*E2.GE.-1.E-6) GO TO 1
      PRINT 24, I
      STOP
1     X(NX)=X(I)
      Y(NX)=Y(I)
      Z(NX)=-E1
      X2(NX)=X2(I)
      Y2(NX)=Y2(I)
      Z2(NX)=-E2
      ITAGI=ITAG(I)
      IF (ITAGI.EQ.0) ITAG(NX)=0
      IF (ITAGI.NE.0) ITAG(NX)=ITAGI+ITI
2     BI(NX)=BI(I)
      N=N*2-N1
      ITI=ITI*2
3     IF (M.LT.M2) GO TO 6
      NXX=LD+1-M1
      DO 5 I=M2,M
      NXX=NXX-1
      NX=NXX-M+M1
      IF (ABS(Z(NXX)).GT.1.E-10) GO TO 4
      PRINT 25, I
      STOP
4     X(NX)=X(NXX)
      Y(NX)=Y(NXX)
      Z(NX)=-Z(NXX)
      T1X(NX)=T1X(NXX)
      T1Y(NX)=T1Y(NXX)
      T1Z(NX)=-T1Z(NXX)
      T2X(NX)=T2X(NXX)
      T2Y(NX)=T2Y(NXX)
      T2Z(NX)=-T2Z(NXX)
      SALP(NX)=-SALP(NXX)
5     BI(NX)=BI(NXX)
```

```
      M=M*2-M1
    6 IF (IY.EQ.0) GO TO 12
C
C     REFLECT ALONG Y AXIS
C
      IF (N.LT.N2) GO TO 9
      DO 8 I=N2,N
      NX=I+N-N1
      E1=Y(I)
      E2=Y2(I)
      IF (ABS(E1)+ABS(E2).GT.1.E-5.AND.E1*E2.GE.-1.E-6) GO TO 7
      PRINT 24, I
      STOP
    7 X(NX)=X(I)
      Y(NX)=-E1
      Z(NX)=Z(I)
      X2(NX)=X2(I)
      Y2(NX)=-E2
      Z2(NX)=Z2(I)
      ITAGI=ITAG(I)
      IF (ITAGI.EQ.0) ITAG(NX)=0
      IF (ITAGI.NE.0) ITAG(NX)=ITAGI+ITI
    8 BI(NX)=BI(I)
      N=N*2-N1
      ITI=ITI*2
    9 IF (M.LT.M2) GO TO 12
      NXX=LD+1-M1
      DO 11 I=M2,M
      NXX=NXX-1
      NX=NXX-M+M1
      IF (ABS(Y(NXX)).GT.1.E-10) GO TO 10
      PRINT 25, I
      STOP
   10 X(NX)=X(NXX)
      Y(NX)=-Y(NXX)
      Z(NX)=Z(NXX)
      T1X(NX)=T1X(NXX)
      T1Y(NX)=-T1Y(NXX)
      T1Z(NX)=T1Z(NXX)
      T2X(NX)=T2X(NXX)
      T2Y(NX)=-T2Y(NXX)
      T2Z(NX)=T2Z(NXX)
      SALP(NX)=-SALP(NXX)
   11 BI(NX)=BI(NXX)
      M=M*2-M1
   12 IF (IX.EQ.0) GO TO 18
C
C     REFLECT ALONG X AXIS
C
      IF (N.LT.N2) GO TO 15
      DO 14 I=N2,N
      NX=I+N-N1
      E1=X(I)
      E2=X2(I)
      IF (ABS(E1)+ABS(E2).GT.1.E-5.AND.E1*E2.GE.-1.E-6) GO TO 13
      PRINT 24, I
      STOP
   13 X(NX)=-E1
      Y(NX)=Y(I)
      Z(NX)=Z(I)
      X2(NX)=-E2
      Y2(NX)=Y2(I)
      Z2(NX)=Z2(I)
      ITAGI=ITAG(I)
```

```
      IF (ITAGI.EQ.0) ITAG(NX)=0
      IF (ITAGI.NE.0) ITAG(NX)=ITAGI+ITI
   14 BI(NX)=BI(I)
      N=N*2-N1
   15 IF (M.LT.M2) GO TO 18
      NXX=LD+1-M1
      DO 17 I=M2,M
      NXX=NXX-1
      NX=NXX-M+M1
      IF (ABS(X(NXX)).GT.1.E-10) GO TO 16
      PRINT 25, I
      STOP
   16 X(NX)=-X(NXX)
      Y(NX)=Y(NXX)
      Z(NX)=Z(NXX)
      T1X(NX)=-T1X(NXX)
      T1Y(NX)=T1Y(NXX)
      T1Z(NX)=T1Z(NXX)
      T2X(NX)=-T2X(NXX)
      T2Y(NX)=T2Y(NXX)
      T2Z(NX)=T2Z(NXX)
      SALP(NX)=-SALP(NXX)
   17 BI(NX)=BI(NXX)
      M=M*2-M1
   18 RETURN
C
C     REPRODUCE STRUCTURE WITH ROTATION TO FORM CYLINDRICAL STRUCTURE
C
   19 FNOP=NOP
      IPSYM=-1
      SAM=6.283185308/FNOP
      CS=COS(SAM)
      SS=SIN(SAM)
      IF (N.LT.N2) GO TO 21
      N=N1+(N-N1)*NOP
      NX=NP+1
      DO 20 I=NX,N
      K=I-NP+N1
      XK=X(K)
      YK=Y(K)
      X(I)=XK*CS-YK*SS
      Y(I)=XK*SS+YK*CS
      Z(I)=Z(K)
      XK=X2(K)
      YK=Y2(K)
      X2(I)=XK*CS-YK*SS
      Y2(I)=XK*SS+YK*CS
      Z2(I)=Z2(K)
      ITAGI=ITAG(K)
      IF (ITAGI.EQ.0) ITAG(I)=0
      IF (ITAGI.NE.0) ITAG(I)=ITAGI+ITI
   20 BI(I)=BI(K)
   21 IF (M.LT.M2) GO TO 23
      M=M1+(M-M1)*NOP
      NX=MP+1
      K=LD+1-M1
      DO 22 I=NX,M
      K=K-1
      J=K-MP+M1
      XK=X(K)
      YK=Y(K)
      X(J)=XK*CS-YK*SS
      Y(J)=XK*SS+YK*CS
      Z(J)=Z(K)
```

```
      XK=T1X(K)                                                         RE 193
      YK=T1Y(K)                                                         RE 194
      T1X(J)=XK*CS-YK*SS                                                RE 195
      T1Y(J)=XK*SS+YK*CS                                                RE 196
      T1Z(J)=T1Z(K)                                                     RE 197
      XK=T2X(K)                                                         RE 198
      YK=T2Y(K)                                                         RE 199
      T2X(J)=XK*CS-YK*SS                                                RE 200
      T2Y(J)=XK*SS+YK*CS                                                RE 201
      T2Z(J)=T2Z(K)                                                     RE 202
      SALP(J)=SALP(K)                                                   RE 203
   22 BI(J)=BI(K)                                                       RE 204
   23 RETURN                                                            RE 205
C                                                                       RE 206
   24 FORMAT (29H GEOMETRY DATA ERROR--SEGMENT,I5,26H LIES IN PLANE OF S RE 207
     1YMMETRY)                                                          RE 208
   25 FORMAT (27H GEOMETRY DATA ERROR--PATCH,I4,26H LIES IN PLANE OF SYM RE 209
     1METRY)                                                            RE 210
      END                                                               RE 211-
```

ROM2

PURPOSE

To numerically integrate over the current distribution on a segment to obtain the field due to the Sommerfeld integral term.

METHOD

ROM2 integrates the product of $\bar{E}_s(\bar{r})$ (see discussion of EFLD) and the current over a segment. Separate integrals are evaluated for current distributions of constant, sin $k(s - s_0)$ and cos $k(s - s_0)$. With three vector components of the field, there are nine integrals evaluated simultaneously and stored in the array SUM. The integration method is the same as that described for subroutine INTX, but loops from one through nine are used at each step.

The parameter DMIN is set in EFLD to

$$\mathrm{DMIN} = 0.01\ [|E_x'|^2 + |E_y'|^2 + |E_z'|^2]^{1/2}$$

where $\bar{E}' = \int_{\mathrm{segment}} [\bar{E}_D(\bar{r}) + \dfrac{k_1^2 - k_2^2}{k_1^2 + k_2^2} \bar{E}_I(\bar{r})]\, ds$.

DMIN is passed to TEST as the lower limit for the denominator in the relative error evaluation to avoid trying to maintain relative accuracy in integrating the Sommerfeld integral when it is much smaller than the other terms.

SYMBOL DICTIONARY

A	= lower limit of integral
B	= upper limit of integral
DMIN	= minimum for denominator in relative error test
DZ	= subinterval size
DZOT	= 0.5 DZ
EP	= tolerance for hitting upper limit

G1, G2, G3, G4, G5	= integrand values at points within the subinterval
N	= number of functions (9)
NM	= minimum subinterval size is (B - A)/NM
NS	= present subinterval size is (B - A)/NS
NT	= counter to control increasing subinterval size
NTS	= larger values retard increasing subinterval size
NX	= maximum subinterval size is (B - A)/NX
RX	= relative error limit
S	= B - A
SUM	= array for integral values
T00, T01, T02, T10, T11, T20	= (see subroutine INTX)
TMAG1, TMAG2	= sum of the magnitudes of the integral contributions for the constant current distribution
Z	= integration variable at left side of subinterval
ZE	= B
ZEND	= upper limit

CONSTANTS

1.E-4	= relative error criterion
65536	= limit for cutting subinterval size

```
      SUBROUTINE ROM2 (A,B,SUM,DMIN)
C
C     FOR THE SOMMERFELD GROUND OPTION, ROM2 INTEGRATES OVER THE SOURCE
C     SEGMENT TO OBTAIN THE TOTAL FIELD DUE TO GROUND.  THE METHOD OF
C     VARIABLE INTERVAL WIDTH ROMBERG INTEGRATION IS USED.  THERE ARE 9
C     FIELD COMPONENTS - THE X, Y, AND Z COMPONENTS DUE TO CONSTANT,
C     SINE, AND COSINE CURRENT DISTRIBUTIONS.
C
      COMPLEX SUM,G1,G2,G3,G4,G5,T00,T01,T10,T02,T11,T20
      DIMENSION SUM(9), G1(9), G2(9), G3(9), G4(9), G5(9), T01(9), T10(9
     1), T20(9)
      DATA NM,NTS,NX,N/65536,4,1,9/,RX/1.E-4/
      Z=A
      ZE=B
      S=B-A
      IF (S.GE.0.) GO TO 1
      PRINT 18
      STOP
1     EP=S/(1.E4*NM)
      ZEND=ZE-EP
      DO 2 I=1,N
2     SUM(I)=(0.,0.)
      NS=NX
      NT=0
      CALL SFLDS (Z,G1)
3     DZ=S/NS
      IF (Z+DZ.LE.ZE) GO TO 4
      DZ=ZE-Z
      IF (DZ.LE.EP) GO TO 17
4     DZOT=DZ*.5
      CALL SFLDS (Z+DZOT,G3)
      CALL SFLDS (Z+DZ,G5)
5     TMAG1=0.
      TMAG2=0.
C
C     EVALUATE 3 POINT ROMBERG RESULT AND TEST CONVERGENCE.
C
      DO 6 I=1,N
      T00=(G1(I)+G5(I))*DZOT
      T01(I)=(T00+DZ*G3(I))*.5
      T10(I)=(4.*T01(I)-T00)/3.
      IF (I.GT.3) GO TO 6
      TR=REAL(T01(I))
      TI=AIMAG(T01(I))
      TMAG1=TMAG1+TR*TR+TI*TI
      TR=REAL(T10(I))
      TI=AIMAG(T10(I))
      TMAG2=TMAG2+TR*TR+TI*TI
6     CONTINUE
      TMAG1=SQRT(TMAG1)
      TMAG2=SQRT(TMAG2)
      CALL TEST(TMAG1,TMAG2,TR,0.,0.,TI,DMIN)
      IF(TR.GT.RX)GO TO 8
      DO 7 I=1,N
7     SUM(I)=SUM(I)+T10(I)
      NT=NT+2
      GO TO 12
8     CALL SFLDS (Z+DZ*.25,G2)
      CALL SFLDS (Z+DZ*.75,G4)
      TMAG1=0.
      TMAG2=0.
C
C     EVALUATE 5 POINT ROMBERG RESULT AND TEST CONVERGENCE.
C
```

```
      DO 9 I=1,N                                                      RO  65
      T02=(T01(I)+DZOT*(G2(I)+G4(I)))*.5                              RO  66
      T11=(4.*T02-T01(I))/3.                                          RO  67
      T20(I)=(16.*T11-T10(I))/15.                                     RO  68
      IF (I.GT.3) GO TO 9                                             RO  69
      TR=REAL(T11)                                                    RO  70
      TI=AIMAG(T11)                                                   RO  71
      TMAG1=TMAG1+TR*TR+TI*TI                                         RO  72
      TR=REAL(T20(I))                                                 RO  73
      TI=AIMAG(T20(I))                                                RO  74
      TMAG2=TMAG2+TR*TR+TI*TI                                         RO  75
    9 CONTINUE                                                        RO  76
      TMAG1=SQRT(TMAG1)                                               RO  77
      TMAG2=SQRT(TMAG2)                                               RO  78
      CALL TEST(TMAG1,TMAG2,TR,0.,0.,TI,DMIN)                         RO  79
      IF(TR.GT.RX)GO TO 14                                            RO  80
   10 DO 11 I=1,N                                                     RO  81
   11 SUM(I)=SUM(I)+T20(I)                                            RO  82
      NT=NT+1                                                         RO  83
   12 Z=Z+DZ                                                          RO  84
      IF (Z.GT.ZEND) GO TO 17                                         RO  85
      DO 13 I=1,N                                                     RO  86
   13 G1(I)=G5(I)                                                     RO  87
      IF (NT.LT.NTS.OR.NS.LE.NX) GO TO 3                              RO  88
      NS=NS/2                                                         RO  89
      NT=1                                                            RO  90
      GO TO 3                                                         RO  91
   14 NT=0                                                            RO  92
      IF (NS.LT.NM) GO TO 15                                          RO  93
      PRINT 19, Z                                                     RO  94
      GO TO 10                                                        RO  95
   15 NS=NS*2                                                         RO  96
      DZ=S/NS                                                         RO  97
      DZOT=DZ*.5                                                      RO  98
      DO 16 I=1,N                                                     RO  99
      G5(I)=G3(I)                                                     RO 100
   16 G3(I)=G2(I)                                                     RO 101
      GO TO 5                                                         RO 102
   17 CONTINUE                                                        RO 103
      RETURN                                                          RO 104
C                                                                     RO 105
   18 FORMAT (30H ERROR - B LESS THAN A IN ROM2)                      RO 106
   19 FORMAT (33H ROM2 -- STEP SIZE LIMITED AT Z =,E12.5)             RO 107
      END                                                             RO 108-
```

SBF

PURPOSE

To evaluate the current expansion function associated with a given segment, returning only that portion on a particular segment.

METHOD

SBF is very similar to routine TBF. Both routines evaluate the current expansion functions. However, while TBF stores the coefficients for each segment on which a given expansion function is non-zero, SBF returns the coefficients for only a single specified segment.

In the call to SBF, I is the segment on which the expansion function is centered. IS is the segment for which the function coefficients A_j, B_j and C_j are requested. These coefficients are returned in AA, BB, CC, respectively.

Refer to TBF for a discussion of the coding and variables. One additional variable in SBF -- JUNE -- is set to -1 or +1 if segment IS is found connected to end 1 or end 2, respectively, of segment I. If I = IS and segment I is not connected to a surface or ground plane, then JUNE is set to 0.

```
      SUBROUTINE SBF (I,IS,AA,BB,CC)
C     COMPUTE COMPONENT OF BASIS FUNCTION I ON SEGMENT IS.
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      DATA PI/3.141592654/,JMAX/30/
      AA=0.
      BB=0.
      CC=0.
      JUNE=0
      JSNO=0
      PP=0.
      JCOX=ICON1(I)
      IF (JCOX.GT.10000) JCOX=I
      JEND=-1
      IEND=-1
      SIG=-1.
      IF (JCOX) 1,11,2
1     JCOX=-JCOX
      GO TO 3
2     SIG=-SIG
      JEND=-JEND
3     JSNO=JSNO+1
      IF (JSNO.GE.JMAX) GO TO 24
      D=PI*SI(JCOX)
      SDH=SIN(D)
      CDH=COS(D)
      SD=2.*SDH*CDH
      IF (D.GT.0.015) GO TO 4
      OMC=4.*D*D
      OMC=((1.3888889E-3*OMC-4.1666666667E-2)*OMC+.5)*OMC
      GO TO 5
4     OMC=1.-CDH*CDH+SDH*SDH
5     AJ=1./(ALOG(1./(PI*BI(JCOX)))-.577215664)
      PP=PP-OMC/SD*AJ
      IF (JCOX.NE.IS) GO TO 6
      AA=AJ/SD*SIG
      BB=AJ/(2.*CDH)
      CC=-AJ/(2.*SDH)*SIG
      JUNE=IEND
6     IF (JCOX.EQ.I) GO TO 9
      IF (JEND.EQ.1) GO TO 7
      JCOX=ICON1(JCOX)
      GO TO 8
7     JCOX=ICON2(JCOX)
8     IF (IABS(JCOX).EQ.I) GO TO 10
      IF (JCOX) 1,24,2
9     IF (JCOX.EQ.IS) BB=-BB
10    IF (IEND.EQ.1) GO TO 12
11    PM=-PP
      PP=0.
      NJUN1=JSNO
      JCOX=ICON2(I)
      IF (JCOX.GT.10000) JCOX=I
      JEND=1
      IEND=1
      SIG=-1.
      IF (JCOX) 1,12,2
12    NJUN2=JSNO-NJUN1
      D=PI*SI(I)
      SDH=SIN(D)
      CDH=COS(D)
      SD=2.*SDH*CDH
      CD=CDH*CDH-SDH*SDH
```

```
      IF (D.GT.0.015) GO TO 13
      OMC=4.*D*D
      OMC=((1.3888889E-3*OMC-4.1666666667E-2)*OMC+.5)*OMC
      GO TO 14
13    OMC=1.-CD
14    AP=1./(ALOG(1./(PI*BI(I)))-.577215664)
      AJ=AP
      IF (NJUN1.EQ.0) GO TO 19
      IF (NJUN2.EQ.0) GO TO 21
      QP=SD*(PM*PP+AJ*AP)+CD*(PM*AP-PP*AJ)
      QM=(AP*OMC-PP*SD)/QP
      QP=-(AJ*OMC+PM*SD)/QP
      IF (JUNE) 15,18,16
15    AA=AA*QM
      BB=BB*QM
      CC=CC*QM
      GO TO 17
16    AA=-AA*QP
      BB=BB*QP
      CC=-CC*QP
17    IF (I.NE.IS) RETURN
18    AA=AA-1.
      BB=BB+(AJ*QM+AP*QP)*SDH/SD
      CC=CC+(AJ*QM-AP*QP)*CDH/SD
      RETURN
19    IF (NJUN2.EQ.0) GO TO 23
      QP=PI*BI(I)
      XXI=QP*QP
      XXI=QP*(1.-.5*XXI)/(1.-XXI)
      QP=-(OMC+XXI*SD)/(SD*(AP+XXI*PP)+CD*(XXI*AP-PP))
      IF (JUNE.NE.1) GO TO 20
      AA=-AA*QP
      BB=BB*QP
      CC=-CC*QP
      IF (I.NE.IS) RETURN
20    AA=AA-1.
      D=CD-XXI*SD
      BB=BB+(SDH+AP*QP*(CDH-XXI*SDH))/D
      CC=CC+(CDH+AP*QP*(SDH+XXI*CDH))/D
      RETURN
21    QM=PI*BI(I)
      XXI=QM*QM
      XXI=QM*(1.-.5*XXI)/(1.-XXI)
      QM=(OMC+XXI*SD)/(SD*(AJ-XXI*PM)+CD*(PM+XXI*AJ))
      IF (JUNE.NE.-1) GO TO 22
      AA=AA*QM
      BB=BB*QM
      CC=CC*QM
      IF (I.NE.IS) RETURN
22    AA=AA-1.
      D=CD-XXI*SD
      BB=BB+(AJ*QM*(CDH-XXI*SDH)-SDH)/D
      CC=CC+(CDH-AJ*QM*(SDH+XXI*CDH))/D
      RETURN
23    AA=-1.
      QP=PI*BI(I)
      XXI=QP*QP
      XXI=QP*(1.-.5*XXI)/(1.-XXI)
      CC=1./(CDH-XXI*SDH)
      RETURN
24    PRINT 25, I
      STOP
C
25    FORMAT (43H SBF - SEGMENT CONNECTION ERROR FOR SEGMENT,I5)
```

SECOND

PURPOSE

To obtain the time in seconds

METHOD

This subroutine acts as an interface of the computer system's time function and the NEC program. The system time function is called, the number is converted to seconds, and returned to the NEC program through the argument of subroutine SECOND. On CDC 6000 series computers, the system time function is SECOND and is called by the NEC program. This subroutine is, therefore, omitted on CDC 6000 computers.

CODE LISTING

```
      SUBROUTINE SECOND (T)                                        SE   1
```

Call system time function and set T equal to time in seconds.

```
      RETURN                                                       SE   9
      END                                                          SE  10-
```

SFLDS

PURPOSE

To evaluate the Sommerfeld-integral field components due to an infinitesimal current element on a segment.

METHOD

The coordinates of the segment are stored in COMMON/DATAJ/. The current element, at a distance T from the center of the segment is located at (XT, YT, ZT). From SL16 to SL42 the ρ, ϕ and z coordinates of the field evaluation point (XO, YO, ZO) are computed in a coordinate system with the z axis passing through the current element and $\phi = 0$ in the direction of the segment reference direction projected on the x,y plane. R2 is as shown in Figure 6 (page 160) and is the same as R1 in Section IV of Part I.

The Sommerfeld-integral field is computed from SL85 to SL111 by giving R2 and θ', with

$$\theta' = \tan^{-1}\left(\frac{z + z'}{\rho}\right),$$

to subroutine INTRP. INTRP returns the quantities in equations 156 through 159 of Part I as

$$ERV = I_\rho^V$$

$$EZV = I_Z^V$$

$$ERH = I_\rho^H$$

$$EPH = I_\phi^H$$

These quantities are then multiplied by $\exp(-jkR_2)/R_2$. The components for a horizontal current element are multiplied by the appropriate factors of $\sin\phi$ or $\cos\phi$ and combined with the components for a vertical current element according to the elevation angle of the segment. Thus lines SL94 to SL96 are the ρ, z and ϕ components of the field of the current element. These are converted to x, y and z components and stored in E(1), E(2) and

E(3). They are also multiplied by sin(kT) and cos(kT) for the sine and cosine current distributions and stored in other elements of E.

When the separation of the source segment and observation point is large enough that the Norton approximation is used for the field, the code from SL49 to SL80 is executed. In this case SFLDS is called directly by EFLD, with T equal to zero, and returns an approximation to the field of the whole segment. The current is lumped at the center for a point source approximation.

GWAVE computes the total field including direct field and the asymptotic approximation of the field due to ground. Since EFLD has already computed

$$\bar{E}_D(\bar{r}) + \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2}\,\bar{E}_I(\bar{r})$$

these terms must be removed from the field computed by GWAVE. The direct field $\bar{E}_D$ is set to zero by setting XX1 to zero before calling GWAVE. The second term is substracted from the field returned by GWAVE from SL59 to SL63. The field components of a vertical (V) and horizontal (H) current element in the direction $\phi = 0$ at the image point are

$$E_\rho^V = (E_R + E_T)\sin\theta\cos\theta$$

$$E_Z^V = E_R\cos^2\theta - E_T\sin^2\theta$$

$$E_\rho^H = (E_R\sin^2\theta - E_T\cos^2\theta)\cos\phi$$

$$E_Z^H = (E_R + E_T)\sin\theta\cos\theta\cos\phi$$

$$E_\phi^H = E_T\sin\phi$$

where

$$E_R = \frac{-j\eta}{4\pi^2}\,\frac{\exp(-jkR_2)}{(R_2/\lambda)^3}\,(1 + jkR_2)$$

$$E_T = \frac{-j\eta}{8\pi^2} \frac{\exp(-jkR_2)}{(R_2/\lambda)^3} (1 - k^2R_2^2 + jkR_2)$$

$$\cos\theta = (z + z')/R_2$$

$$\sin\theta = \rho/R_2$$

and current moment, $I\ell/\lambda^2 = 1$.

The sin ϕ and cos ϕ factors are omitted to match the quantities returned by GWAVE. Also, the fields of the horizontal current are reversed since the image of the source is in the direction $\phi = 180$ degrees. These quantities are multiplied by FRATI and subtracted from the fields returned by GWAVE.

The total field, in x, y and z components, is stored from SL70 to SL72. S is the length of the segment in wavelengths. Hence it is $I\ell/\lambda^2$ when $I/\lambda = 1$. The current moment for a sine distribution is zero and for a cosine distribution is $\sin(\pi S)/\pi$.

SYMBOL DICTIONARY

CPH	$= \cos\phi$
E	= array for returning field components
EPH	$= E_\phi^H$ or I_ϕ^H
ER	$= E_R$
ERH	$= E_\rho^H$ or I_ρ^H
ERV	$= E_\rho^V$ or I_ρ^V
ET	$= E_T$
EZH	$= E_Z^H$ or I_Z^H
EZV	$= E_Z^V$ or I_Z^V
FRATI	$= (k_1^2 - k_2^2)/(k_1^2 + k_2^2)$
HRH	$= E_\rho^H$ for image of source current element

SFLDS

HRV	=	E_ρ^V
HZV	=	H_Z^V
PHX	=	x component of $\hat{\phi}$
PHY	=	y component of $\hat{\phi}$
PI	=	π
POT	=	$\pi/2$
R1	=	direct distance to source (set to arbitrary value)
R2	=	distance to image
R2S	=	$(R2)^2$
RHØ	=	ρ
RHS	=	ρ^2
RHX	=	x component of ρ
RHY	=	y component of ρ
RK	=	kR_2
SFAC	=	value of current or current moment
SPH	=	$\sin \phi$
T	=	distance from center of segment to current element
THET	=	θ'
TP	=	2π
XT, YT, ZT	=	coordinates of current element
ZPHS	=	$(z + z')^2$

CONSTANTS

1.570796327	=	$\pi/2$
3.141592654	=	π
6.283185308	=	2π

```
      SUBROUTINE SFLDS (T,E)
C
C     SFLDX RETURNS THE FIELD DUE TO GROUND FOR A CURRENT ELEMENT ON
C     THE SOURCE SEGMENT AT T RELATIVE TO THE SEGMENT CENTER.
C
      COMPLEX E,ERV,EZV,ERH,EZH,EPH,T1,EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,E
     1ZC,XX1,XX2,U,U2,ZRATI,ZRATI2,FRATI,ER,ET,HRV,HZV,HRH
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      COMMON /INCOM/ XO,YO,ZO,SN,XSN,YSN,ISNOR
      COMMON /GWAV/ U,U2,XX1,XX2,R1,R2,ZMH,ZPH
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      DIMENSION E(9)
      DATA PI/3.141592654/,TP/6.283185308/,POT/1.570796327/
      XT=XJ+T*CABJ
      YT=YJ+T*SABJ
      ZT=ZJ+T*SALPJ
      RHX=XO-XT
      RHY=YO-YT
      RHS=RHX*RHX+RHY*RHY
      RHO=SQRT(RHS)
      IF (RHO.GT.0.) GO TO 1
      RHX=1.
      RHY=0.
      PHX=0.
      PHY=1.
      GO TO 2
1     RHX=RHX/RHO
      RHY=RHY/RHO
      PHX=-RHY
      PHY=RHX
2     CPH=RHX*XSN+RHY*YSN
      SPH=RHY*XSN-RHX*YSN
      IF (ABS(CPH).LT.1.E-10) CPH=0.
      IF (ABS(SPH).LT.1.E-10) SPH=0.
      ZPH=ZO+ZT
      ZPHS=ZPH*ZPH
      R2S=RHS+ZPHS
      R2=SQRT(R2S)
      RK=R2*TP
      XX2=CMPLX(COS(RK),-SIN(RK))
      IF (ISNOR.EQ.1) GO TO 3
C
C     USE NORTON APPROXIMATION FOR FIELD DUE TO GROUND.  CURRENT IS
C     LUMPED AT SEGMENT CENTER WITH CURRENT MOMENT FOR CONSTANT, SINE,
C     OR COSINE DISTRIBUTION.
C
      ZMH=1.
      R1=1.
      XX1=0.
      CALL GWAVE (ERV,EZV,ERH,EZH,EPH)
      ET=-(0.,4.77134)*FRATI*XX2/(R2S*R2)
      ER=2.*ET*CMPLX(1.,RK)
      ET=ET*CMPLX(1.-RK*RK,RK)
      HRV=(ER+ET)*RHO*ZPH/R2S
      HZV=(ZPHS*ER-RHS*ET)/R2S
      HRH=(RHS*ER-ZPHS*ET)/R2S
      ERV=ERV-HRV
      EZV=EZV-HZV
      ERH=ERH+HRH
      EZH=EZH+HRV
      EPH=EPH+ET
      ERV=ERV*SALPJ
```

```
      EZV=EZV*SALPJ
      ERH=ERH*SN*CPH
      EZH=EZH*SN*CPH
      EPH=EPH*SN*SPH
      ERH=ERV+ERH
      E(1)=(ERH*RHX+EPH*PHX)*S
      E(2)=(ERH*RHY+EPH*PHY)*S
      E(3)=(EZV+EZH)*S
      E(4)=0.
      E(5)=0.
      E(6)=0.
      SFAC=PI*S
      SFAC=SIN(SFAC)/SFAC
      E(7)=E(1)*SFAC
      E(8)=E(2)*SFAC
      E(9)=E(3)*SFAC
      RETURN
C
      INTERPOLATE IN SOMMERFELD FIELD TABLES
C
3     IF (RHO.LT.1.E-12) GO TO 4
      THET=ATAN(ZPH/RHO)
      GO TO 5
4     THET=POT
5     CALL INTRP (R2,THET,ERV,EZV,ERH,EPH)
C     COMBINE VERTICAL AND HORIZONTAL COMPONENTS AND CONVERT TO X,Y,Z
C     COMPONENTS.  MULTIPLY BY EXP(-JKR)/R.
      XX2=XX2/R2
      SFAC=SN*CPH
      ERH=XX2*(SALPJ*ERV+SFAC*ERH)
      EZH=XX2*(SALPJ*EZV-SFAC*ERV)
      EPH=SN*SPH*XX2*EPH
C     X,Y,Z FIELDS FOR CONSTANT CURRENT
      E(1)=ERH*RHX+EPH*PHX
      E(2)=ERH*RHY+EPH*PHY
      E(3)=EZH
      RK=TP*T
C     X,Y,Z FIELDS FOR SINE CURRENT
      SFAC=SIN(RK)
      E(4)=E(1)*SFAC
      E(5)=E(2)*SFAC
      E(6)=E(3)*SFAC
C     X,Y,Z FIELDS FOR COSINE CURRENT
      SFAC=COS(RK)
      E(7)=E(1)*SFAC
      E(8)=E(2)*SFAC
      E(9)=E(3)*SFAC
      RETURN
      END
```

SOLGF

PURPOSE

To solve for the basis function amplitudes in the NGF procedure.

METHOD

The operations performed here are described in the NGF overview in Section VI. SOLGF is called for either a NGF solution or a normal solution. For the normal solution, or for a NGF solution when no new segments or patches have been added, the solution is obtained by calling SOLVES at SF14. Otherwise, the rest of the code is executed.

The excitation vector XY is filled in the subroutine ETMNS in the order

1. E on NGF segments (N1 elements)
2. E on new segments (N - N1 elements)
3. H on NGF patches (2M1 elements)
4. H on new patches (2M - 2M1 elements)

From SF18 to SF29 this vector is put in the order

$$\left.\begin{array}{l}\text{1. E on NGF segments}\\ \text{2. H on NGF patches}\end{array}\right\} E_1$$

$$\left.\begin{array}{l}\text{3. E on new segments}\\ \text{4. H on new patches}\end{array}\right\} E_2$$

to conform to the matrix structure. From SF30 to SF36, zeros are stored in XY in the locations opposite the rows of the C' matrix. Line SF37 then computes $A^{-1}E_1$ storing it in place of E_1.

SF41 to SF52 computes $E_2 - C\,A^{-1}E_1$ and stores it in palce of E_2. Matrix C is read from file 15 if necessary to form the product with $A^{-1}E_1$. From SF55 to SF80

$$I_2 = [D - CA^{-1}B]^{-1}[E_2 - CA^{-1}E_1]$$

is computed in the original location of E_2. If ICASX is 4 the block parameters for the primary matrix are temporarily changed to those of $D - CA^{-1}B$ so that LTSOLV, which uses the primary block parameters, can perform the solution procedure. From SF84 to SF95

$$I_1 = A^{-1}E_1 - (A^{-1}B)I_2$$

is computed. The reordering step at the beginning of SOLGF is then reversed from SF98 to SF107 to put the solution vector in the order

1. amplitudes of NGF basis functions
2. amplitudes of new basis functions
3. NGF patch currents
4. new patch currents
5. amplitudes of modified basis functions for NGF segments that connect to new segments
6. meaningless values associated with B'_{ss}

Finally, from SF109 to SF113 the amplitudes of the modified basis functions are stored in place of the NGF basis functions that were set to zero.

SYMBOL DICTIONARY

A	= array for matrix A_F
B	= array starting just after A in CM (used for factoring $D - CA^{-1}B$ for ICASX = 2, 3 or 4)
C	= array for matrix C
D	= array used for factoring $D - CA^{-1}B$ when ICASX = 1
ICASS	= saved value of ICASE
IFL	= file in which blocks of A_F are stored in descending order (ascending order is always on 13)
IP	= array of pivot element indices
M	= number of patches

M1	= number of patches in NGF
MP	= number of patches in one symmetric section of the NGF structure
N	= number of segments
N1	= number of segments in NGF
N1C	= number of unknowns in NGF (N1 + 2M1)
N2	= N1 + 1
N2C	= number of new unknowns (order of D)
NBLSYS	= saved value of NBLSYM
NEQ	= total number of unknowns (NGF and new)
NEQS	= number of columns in B'_{sw} and B'_{ss}
NLSYS	= saved value of NLSYM
NP	= number of segments in a symmetric section of the NGF structure
NPSYS	= saved value of NPSYM
SUM	= summation variable for matrix products
XY	= excitation and solution vector

```
      SUBROUTINE SOLGF (A,B,C,D,XY,IP,NP,N1,N,MP,M1,M,N1C,N2C)
C     SOLVE FOR CURRENT IN N.G.F. PROCEDURE
      COMPLEX A,B,C,D,SUM,XY,Y
      COMMON /SCRATM/ Y(600)
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP
     1CON(10),NPCON
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      DIMENSION B(N1C,1), C(N1C,1), D(N2C,1), IP(1), XY(1)
      IFL=14
      IF (ICASX.GT.0) IFL=13
      IF (N2C.GT.0) GO TO 1
C     NORMAL SOLUTION.  NOT N.G.F.
      CALL SOLVES (A,IP,XY,N1C,1,NP,N,MP,M,13,IFL)
      GO TO 22
1     IF (N1.EQ.N.OR.M1.EQ.0) GO TO 5
C     REORDER EXCITATION ARRAY
      N2=N1+1
      JJ=N+1
      NPM=N+2*M1
      DO 2 I=N2,NPM
2     Y(I)=XY(I)
      J=N1
      DO 3 I=JJ,NPM
      J=J+1
3     XY(J)=Y(I)
      DO 4 I=N2,N
      J=J+1
4     XY(J)=Y(I)
5     NEQS=NSCON+2*NPCON
      IF (NEQS.EQ.0) GO TO 7
      NEQ=N1C+N2C
      NEQS=NEQ-NEQS+1
C     COMPUTE INV(A)E1
      DO 6 I=NEQS,NEQ
6     XY(I)=(0.,0.)
7     CALL SOLVES (A,IP,XY,N1C,1,NP,N1,MP,M1,13,IFL)
      NI=0
      NPB=NPBL
C     COMPUTE E2-C(INV(A)E1)
      DO 10 JJ=1,NBBL
      IF (JJ.EQ.NBBL) NPB=NLBL
      IF (ICASX.GT.1) READ (15) ((C(I,J),I=1,N1C),J=1,NPB)
      II=N1C+NI
      DO 9 I=1,NPB
      SUM=(0.,0.)
      DO 8 J=1,N1C
8     SUM=SUM+C(J,I)*XY(J)
      J=II+I
9     XY(J)=XY(J)-SUM
10    NI=NI+NPBL
      REWIND 15
      JJ=N1C+1
C     COMPUTE INV(D)(E2-C(INV(A)E1)) = I2
      IF (ICASX.GT.1) GO TO 11
      CALL SOLVE (N2C,D,IP(JJ),XY(JJ),N2C)
      GO TO 13
11    IF (ICASX.EQ.4) GO TO 12
      NI=N2C*N2C
      READ (11) (B(J,1),J=1,NI)
      REWIND 11
      CALL SOLVE (N2C,B,IP(JJ),XY(JJ),N2C)
      GO TO 13
12    NBLSYS=NBLSYM
```

```
      NPSYS=NPSYM
      NLSYS=NLSYM
      ICASS=ICASE
      NBLSYM=NBBL
      NPSYM=NPBL
      NLSYM=NLBL
      ICASE=3
      REWIND 11
      REWIND 16
      CALL LTSOLV (B,N2C,IP(JJ),XY(JJ),N2C,1,11,16)
      REWIND 11
      REWIND 16
      NBLSYM=NBLSYS
      NPSYM=NPSYS
      NLSYM=NLSYS
      ICASE=ICASS
13    NI=0
      NPB=NPBL
C     COMPUTE INV(A)E1-(INV(A)B)I2 = I1
      DO 16 JJ=1,NBBL
      IF (JJ.EQ.NBBL) NPB=NLBL
      IF (ICASX.GT.1) READ (14) ((B(I,J),I=1,N1C),J=1,NPB)
      II=N1C+NI
      DO 15 I=1,N1C
      SUM=(0.,0.)
      DO 14 J=1,NPB
      JP=II+J
14    SUM=SUM+B(I,J)*XY(JP)
15    XY(I)=XY(I)-SUM
16    NI=NI+NPBL
      REWIND 14
      IF (N1.EQ.N.OR.M1.EQ.0) GO TO 20
C     REORDER CURRENT ARRAY
      DO 17 I=N2,NPM
17    Y(I)=XY(I)
      JJ=N1C+1
      J=N1
      DO 18 I=JJ,NPM
      J=J+1
18    XY(J)=Y(I)
      DO 19 I=N2,N1C
      J=J+1
19    XY(J)=Y(I)
20    IF (NSCON.EQ.0) GO TO 22
      J=NEQS-1
      DO 21 I=1,NSCON
      J=J+1
      JJ=ISCON(I)
21    XY(JJ)=XY(J)
22    RETURN
      END
```

SOLVE

PURPOSE

To solve the system $LUx = B$, where L is a lower triangular matrix with ones on the diagonal, U is an upper triangular matrix, and B is the right-hand side vector (RHS).

METHOD

The algorithm used is described on pages 409-415 of ref. 1. The solution of the matrix equation $LUx = B$ is found by first solving

$$Ly = B, \tag{3}$$

and then

$$Ux = y, \tag{4}$$

since

$$LUx = Ly = B .$$

The solution of equations (3) and (4) is straightforward since the matrices are both triangular. The solution of equation (3) can be written

$$y_i = \frac{1}{\ell_{ii}}\left(b_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j\right) \quad i = 1, \ldots, n .$$

Equation (4) can be written similarly.

The L and U matrices are both supplied by the subroutine FACTR and are stored in the matrix A; the 1's on the diagonal of L are suppressed. Care must be exercised in the solution, since rows were interchanged during factorization, and this necessitates rearranging the RHS vector; furthermore, the L matrix itself is not completely rearranged. The information pertinent to the row rearrangements has been stored by FACTR in an integer array (IP), and it is used in the computations. The final solution of the equations is overwritten on the input RHS vector B.

The only differences between the coding in SOLVE and the coding suggested in ref. 1 are: (1) double precision variables are not used for the accumulation of sums, since, for the size of matrices anticipated in core, the computer word length is sufficient, and (2) the transposes of the L and U matrices are supplied in A by FACTR. Thus, the row and column indices used in the routine are reversed to account for this transposition.

CODING

S015 - S025 The solution for y in equation (3).

S029 - S039 The solution for x in equation (4) and the storage of the solution in B.

SYMBOL DICTIONARY

A = array contains the input L and U matrices

B = array contains the input RHS and is overwritten with the solution

I = DO loop index

IP = array contains row positioning information

IP1 = I + 1

J = DO loop index

K = DO loop index

N = order of the matrix being solved

NDIM = dimension of the array where the matrix is stored NDIM $\geq$ N

PI = intermediate integer

SUM = intermediate variable

Y = *scratch vector*

SOLVE

```
      SUBROUTINE SOLVE (N,A,IP,B,NDIM)
C
C     SUBROUTINE TO SOLVE THE MATRIX EQUATION LU*X=B WHERE L IS A UNIT
C     LOWER TRIANGULAR MATRIX AND U IS AN UPPER TRIANGULAR MATRIX BOTH
C     OF WHICH ARE STORED IN A.  THE RHS VECTOR B IS INPUT AND THE
C     SOLUTION IS RETURNED THROUGH VECTOR B.    (MATRIX TRANSPOSED.
C
      COMPLEX A,B,Y,SUM
      INTEGER PI
      COMMON /SCRATM/ Y(600)
      DIMENSION A(NDIM,NDIM), IP(NDIM), B(NDIM)
C
C     FORWARD SUBSTITUTION
C
      DO 3 I=1,N
      PI=IP(I)
      Y(I)=B(PI)
      B(PI)=B(I)
      IP1=I+1
      IF (IP1.GT.N) GO TO 2
      DO 1 J=IP1,N
      B(J)=B(J)-A(I,J)*Y(I)
1     CONTINUE
2     CONTINUE
3     CONTINUE
C
C     BACKWARD SUBSTITUTION
C
      DO 6 K=1,N
      I=N-K+1
      SUM=(0.,0.)
      IP1=I+1
      IF (IP1.GT.N) GO TO 5
      DO 4 J=IP1,N
      SUM=SUM+A(J,I)*B(J)
4     CONTINUE
5     CONTINUE
      B(I)=(Y(I)-SUM)/A(I,I)
6     CONTINUE
      RETURN
      END
```

SOLVES

PURPOSE

To control solution of the matrix equation, including transforming and reordering the solution vector.

METHOD

When SOLVES is called, the array B contains the excitation computed by subroutines ETMNS or NETWK. The exciting electric field on all segments is stored first in B, followed by the magnetic fields on all patches. In the case of a symmetric structure, however, the matrix is filled with the coefficients of all segment and patch equations in the first symmetric sector occurring first. These are followed by the coefficients for successive sectors in the same order. This order is required for the solution procedure for symmetric structures described in section III-5 of Part I. For the case of a symmetric structure with both segments and patches, SOLVES first rearranges the excitation coefficients in array B to correspond to the order of the matrix coefficients.

For symmetric structures, SOLVES then computes the transforms of the subvectors in B according to equation (88) of Part I. Subroutine SOLVE or LTSOLV is then called to compute the solution or solution subvectors. The procedure is selected by the parameter ICASE as follows.

1 No symmetry, matrix in core. SOLVE is called for the solution.
2 Symmetry, matrix in core. SOLVE is called for each subvector.
3 No symmetry, matrix out of core. LTSOLV is called for the solution.
4 Symmetry, complete matrix does not fit in core but submatrices do. SOLVE is called for each subvector after first reading the appropriate submatrix from file IFL1.
5 Symmetry, submatrices do not fit in core. LTSOLV is called for each subvector.

SOLVES then computes the total current by inverse transforming the subvectors by equation (115) of Part I. For a symmetric structure with segments and patches, SOLVES then rearranges the solution in array B to put all segment currents first, followed by all patch currents, which is the order of the original excitation coefficients.

SOLVES

Multiple right-hand-side vectors (NRH) may be processed simultaneously at each step in SOLVES. This reduces the time spent reading files when LTSOLV is called, and is used in computing $A^{-1}B$ in the NGF procedure.

CODING

SS22 - SS39 Rearrange excitation coefficients.

SS43 - SS56 Transform subvectors.

SS63 - SS75 Solve for each subvector.

SS81 - SS94 Inverse transform subvectors.

SS96 - SS113 Rearrange solution coefficients.

SYMBOL DICTIONARY

A = array set aside for in-core matrix storage, i.e., factored matrices

B = right-hand side; the solution is overwritten on this array also

FNOP = decimal form of NOP

FNORM = 1/FNOP

IFL1 = file with matrix blocks in normal order

IFL2 = file with matrix blocks in reversed order

IP = array containing positioning data used in SOLVE

M = number of patches

MP = number of patches in a symmetric sector

N = number of segments

NCOL = number of columns in array A

NEQ = order of complete matrix

NOP = number of symmetric sectors

NP = number of segments in a symmetric sector

NPEQ = order of a submatrix

NRH = number of right-hand-side vectors in B

NROW = number of rows in A

SSX = array containing the coefficients S_{ik} in equation (89) of Part I

SUM = summation variable

Y = scratch vector

```
      SUBROUTINE SOLVES (A,IP,B,NEQ,NRH,NP,N,MP,M,IFL1,IFL2)
C
C     SUBROUTINE SOLVES, FOR SYMMETRIC STRUCTURES, HANDLES THE
C     TRANSFORMATION OF THE RIGHT HAND SIDE VECTOR AND SOLUTION OF THE
C     MATRIX EQ.
C
      COMPLEX A,B,Y,SUM,SSX
      COMMON /SMAT/ SSX(16,16)
      COMMON /SCRATM/ Y(600)
      COMMON /MATPAR/ ICASE,NBLOKS,NPBLK,NLAST,NBLSYM,NPSYM,NLSYM,IMAT,I
     1CASX,NBBX,NPBX,NLBX,NBBL,NPBL,NLBL
      DIMENSION A(1), IP(1), B(NEQ,NRH)
      NPEQ=NP+2*MP
      NOP=NEQ/NPEQ
      FNOP=NOP
      FNORM=1./FNOP
      NROW=NEQ
      IF (ICASE.GT.3) NROW=NPEQ
      IF (NOP.EQ.1) GO TO 11
      DO 10 IC=1,NRH
      IF (N.EQ.0.OR.M.EQ.0) GO TO 6
      DO 1 I=1,NEQ
1     Y(I)=B(I,IC)
      KK=2*MP
      IA=NP
      IB=N
      J=NP
      DO 5 K=1,NOP
      IF (K.EQ.1) GO TO 3
      DO 2 I=1,NP
      IA=IA+1
      J=J+1
2     B(J,IC)=Y(IA)
      IF (K.EQ.NOP) GO TO 5
3     DO 4 I=1,KK
      IB=IB+1
      J=J+1
4     B(J,IC)=Y(IB)
5     CONTINUE
C
C     TRANSFORM MATRIX EQ. RHS VECTOR ACCORDING TO SYMMETRY MODES
C
6     DO 10 I=1,NPEQ
      DO 7 K=1,NOP
      IA=I+(K-1)*NPEQ
7     Y(K)=B(IA,IC)
      SUM=Y(1)
      DO 8 K=2,NOP
8     SUM=SUM+Y(K)
      B(I,IC)=SUM*FNORM
      DO 10 K=2,NOP
      IA=I+(K-1)*NPEQ
      SUM=Y(1)
      DO 9 J=2,NOP
9     SUM=SUM+Y(J)*CONJG(SSX(K,J))
10    B(IA;IC)=SUM*FNORM
11    IF (ICASE.LT.3) GO TO 12
      REWIND IFL1
      REWIND IFL2
C
C     SOLVE EACH MODE EQUATION
C
12    DO 16 KK=1,NOP
      IA=(KK-1)*NPEQ+1
```

```
      IB=IA
      IF (ICASE.NE.4) GO TO 13
      I=NPEQ*NPEQ
      READ (IFL1) (A(J),J=1,I)
      IB=1
   13 IF (ICASE.EQ.3.OR.ICASE.EQ.5) GO TO 15
      DO 14 IC=1,NRH
   14 CALL SOLVE (NPEQ,A(IB),IP(IA),B(IA,IC),NROW)
      GO TO 16
   15 CALL LTSOLV (A,NPEQ,IP(IA),B(IA,1),NEQ,NRH,IFL1,IFL2)
   16 CONTINUE
      IF (NOP.EQ.1) RETURN
C
C     INVERSE TRANSFORM THE MODE SOLUTIONS
C
      DO 26 IC=1,NRH
      DO 20 I=1,NPEQ
      DO 17 K=1,NOP
      IA=I+(K-1)*NPEQ
   17 Y(K)=B(IA,IC)
      SUM=Y(1)
      DO 18 K=2,NOP
   18 SUM=SUM+Y(K)
      B(I,IC)=SUM
      DO 20 K=2,NOP
      IA=I+(K-1)*NPEQ
      SUM=Y(1)
      DO 19 J=2,NOP
   19 SUM=SUM+Y(J)*SSX(K,J)
   20 B(IA,IC)=SUM
      IF (N.EQ.0.OR.M.EQ.0) GO TO 26
      DO 21 I=1,NEQ
   21 Y(I)=B(I,IC)
      KK=2*MP
      IA=NP
      IB=N
      J=NP
      DO 25 K=1,NOP
      IF (K.EQ.1) GO TO 23
      DO 22 I=1,NP
      IA=IA+1
      J=J+1
   22 B(IA,IC)=Y(J)
      IF (K.EQ.NOP) GO TO 25
   23 DO 24 I=1,KK
      IB=IB+1
      J=J+1
   24 B(IB,IC)=Y(J)
   25 CONTINUE
   26 CONTINUE
      RETURN
      END
```

TBF

PURPOSE

To evaluate the current expansion function associated with a given segment.

METHOD

The current expansion function is described in section III-1 of Part I. The parameter I is the number of the segment on which the function is centered. On segment I and on all segments connected to either end of segment I, the function has the form

$$f_j(s) = A_j + B_j \sin [k(s - s_j)] + C_j \cos [k(s - s_j)] ,$$

where j is the segment number. TBF locates all connected segments and stores the segment numbers, j, in JCO in COMMON/SEGJ/. It computes A_j, B_j, and C_j and stores them in AX, BX, and CX, respectively, in the same location as was used in JCO. A_j, B_j, and C_j for j = I are stored last in the arrays.

If ICAP = 0, the function goes to zero at an end of segment I to which no other segment or surface is connected. If ICAP ≠ 0, the function has a non-zero value at a free end, allowing for the current onto the wire end cap.

CODING

Equations and symbols refer to Part I.

TB9 - TB55 This code forms a loop that locates all segments connected to the ends of segment I, first for end 1 (IEND = -1) and then for end 2 (IEND = 1).

TB9 - TB16 Parameters are initialized to start search for segments connected to end 1 of segment I.

TB34 PP = P_i^- for end 1 of segment I or P_i^+ for end 2 of segment I.

TB35 - TB37 Equations (43) to (48) of Part I evaluated except for $Q_i^\pm$:

AX(JSNO) = $A_j^\pm/Q_i^\pm$

BX(JSNO) = $B_j^\pm/Q_i^\pm$

CX(JSNO) = $C_j^\pm/Q_i^\pm$

JCO(JSNO) = j

TB38 Exit from loop if segment I is connected to a surface or ground plane. Segment I will occur in COMMON/SEGJ/ twice

	in this case, once for the center of the expansion function on segment I and once for the part of the function extending onto the image of segment I in the surface. Line TB45 changes the sign of $B_j^{\pm}$ for the image term. The sum of the two parts of the function on segment I then has zero derivative at the end connected to the surface.
TB39 - TB42	Check appropriate end of segment j to determine whether it shows a connection to segment I (end of search) or connection to another segment (multiple junction).
TB44	Continue search for connected segments (multiple junction).
TB46	Exit from loop after finishing search for both ends of segment I.
TB47 - TB55	Store values for end 1 of segment I and initialize for end 2. Then return to previous loop.
TB59 - TB70	Evaluate functions of segment length and radius for segment I. For $k\Delta < 0.03$, a series is used for $1 - \cos k\Delta$, where Δ = segment length.
TB73 - TB86	Final calculations if neither end of segment I is a free end.
TB89 - TB102	Final calculations for free end on end 1 of segment I.
TB104 - TB117	Final calculations for free end on end 2 of segment I.
TB119 - TB126	Final calculations for free ends on both ends of segment I.
TB128	$A_j = -1$ for $j = I$ in all cases.

SYMBOL DICTIONARY

AJ = a_j^-

AP = a_j^+

CD = $\cos k\Delta_j$

CDH = $\cos(k\Delta_j/2)$

D = $k\Delta_j/2$ or $\cos k\Delta_i - X_i \sin k\Delta_i$

ICAP = flag to determine whether the function goes to zero at a free end

IEND = -1 during calculations for end 1 of segment I and +1 for end 2.

JCOX = connection index

JEND = -1 if end 1 of a segment is connected to segment I, +1 if end 2 is connected to segment I.

JMAX = maximum number of segments allowed in the expansion function. This includes segment I and all segments connected to either end.

JSNOP = JSN + 1

NJUN1 = N^-

NJUN2 = N^+

OMC = $1 - \cos k\Delta_j$

PI = π

PM = P_i^-

PP = P_i^+

QM = Q_i^-

QP = Q_i^+

SD = $\sin k\Delta_j$

SDH = $\sin (k\Delta_j/2)$

SIG = sign for calculation of A_j and C_j

XXI = $J_1(ka_i)/J_0(ka_i)$ (small argument series used for Bessel functions)

CONSTANTS

0.577215664 = Eulers constant

0.015 = 0.03/2

1.3888889E-3 = 1/720

3.141592654 = π

4.1666666667E-2 = 1/24

TBF

```
      SUBROUTINE TBF (I,ICAP)
C     COMPUTE BASIS FUNCTION I
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP
     1CON(10),NPCON
      DATA PI/3.141592654/,JMAX/30/
      JSNO=0
      PP=0.
      JCOX=ICON1(I)
      IF (JCOX.GT.10000) JCOX=I
      JEND=-1
      IEND=-1
      SIG=-1.
      IF (JCOX) 1,10,2
1     JCOX=-JCOX
      GO TO 3
2     SIG=-SIG
      JEND=-JEND
3     JSNO=JSNO+1
      IF (JSNO.GE.JMAX) GO TO 28
      JCO(JSNO)=JCOX
      D=PI*SI(JCOX)
      SDH=SIN(D)
      CDH=COS(D)
      SD=2.*SDH*CDH
      IF (D.GT.0.015) GO TO 4
      OMC=4.*D*D
      OMC=((1.3888889E-3*OMC-4.1666666667E-2)*OMC+.5)*OMC
      GO TO 5
4     OMC=1.-CDH*CDH+SDH*SDH
5     AJ=1./(ALOG(1./(PI*BI(JCOX)))-.577215664)
      PP=PP-OMC/SD*AJ
      AX(JSNO)=AJ/SD*SIG
      BX(JSNO)=AJ/(2.*CDH)
      CX(JSNO)=-AJ/(2.*SDH)*SIG
      IF (JCOX.EQ.I) GO TO 8
      IF (JEND.EQ.1) GO TO 6
      JCOX=ICON1(JCOX)
      GO TO 7
6     JCOX=ICON2(JCOX)
7     IF (IABS(JCOX).EQ.I) GO TO 9
      IF (JCOX) 1,28,2
8     BX(JSNO)=-BX(JSNO)
9     IF (IEND.EQ.1) GO TO 11
10    PM=-PP
      PP=0.
      NJUN1=JSNO
      JCOX=ICON2(I)
      IF (JCOX.GT.10000) JCOX=I
      JEND=1
      IEND=1
      SIG=-1.
      IF (JCOX) 1,11,2
11    NJUN2=JSNO-NJUN1
      JSNOP=JSNO+1
      JCO(JSNOP)=I
      D=PI*SI(I)
      SDH=SIN(D)
      CDH=COS(D)
      SD=2.*SDH*CDH
      CD=CDH*CDH-SDH*SDH
      IF (D.GT.0.015) GO TO 12
```

```
      OMC=4.*D*D
      OMC=((1.3888889E-3*OMC-4.1666666667E-2)*OMC+.5)*OMC
      GO TO 13
   12 OMC=1.-CD
   13 AP=1./(ALOG(1./(PI*BI(I)))-.577215664)
      AJ=AP
      IF (NJUN1.EQ.0) GO TO 16
      IF (NJUN2.EQ.0) GO TO 20
      QP=SD*(PM*PP+AJ*AP)+CD*(PM*AP-PP*AJ)
      QM=(AP*OMC-PP*SD)/QP
      QP=-(AJ*OMC+PM*SD)/QP
      BX(JSNOP)=(AJ*QM+AP*QP)*SDH/SD
      CX(JSNOP)=(AJ*QM-AP*QP)*CDH/SD
      DO 14 IEND=1,NJUN1
      AX(IEND)=AX(IEND)*QM
      BX(IEND)=BX(IEND)*QM
   14 CX(IEND)=CX(IEND)*QM
      JEND=NJUN1+1
      DO 15 IEND=JEND,JSNO
      AX(IEND)=-AX(IEND)*QP
      BX(IEND)=BX(IEND)*QP
   15 CX(IEND)=-CX(IEND)*QP
      GO TO 27
   16 IF (NJUN2.EQ.0) GO TO 24
      IF (ICAP.NE.0) GO TO 17
      XXI=0.
      GO TO 18
   17 QP=PI*BI(I)
      XXI=QP*QP
      XXI=QP*(1.-.5*XXI)/(1.-XXI)
   18 QP=-(OMC+XXI*SD)/(SD*(AP+XXI*PP)+CD*(XXI*AP-PP))
      D=CD-XXI*SD
      BX(JSNOP)=(SDH+AP*QP*(CDH-XXI*SDH))/D
      CX(JSNOP)=(CDH+AP*QP*(SDH+XXI*CDH))/D
      DO 19 IEND=1,NJUN2
      AX(IEND)=-AX(IEND)*QP
      BX(IEND)=BX(IEND)*QP
   19 CX(IEND)=-CX(IEND)*QP
      GO TO 27
   20 IF (ICAP.NE.0) GO TO 21
      XXI=0.
      GO TO 22
   21 QM=PI*BI(I)
      XXI=QM*QM
      XXI=QM*(1.-.5*XXI)/(1.-XXI)
   22 QM=(OMC+XXI*SD)/(SD*(AJ-XXI*PM)+CD*(PM+XXI*AJ))
      D=CD-XXI*SD
      BX(JSNOP)=(AJ*QM*(CDH+XXI*SDH)-SDH)/D
      CX(JSNOP)=(CDH-AJ*QM*(SDH+XXI*CDH))/D
      DO 23 IEND=1,NJUN1
      AX(IEND)=AX(IEND)*QM
      BX(IEND)=BX(IEND)*QM
   23 CX(IEND)=CX(IEND)*QM
      GO TO 27
   24 BX(JSNOP)=0.
      IF (ICAP.NE.0) GO TO 25
      XXI=0.
      GO TO 26
   25 QP=PI*BI(I)
      XXI=QP*QP
      XXI=QP*(1.-.5*XXI)/(1.-XXI)
   26 CX(JSNOP)=1./(CDH-XXI*SDH)
   27 JSNO=JSNOP
      AX(JSNO)=-1.
```

```
          RETURN                                                            TB 129
      28  PRINT 29, I                                                       TB 130
          STOP                                                              TB 131
C                                                                           TB 132
      29  FORMAT (43H TBF - SEGMENT CONNECTION ERROR FOR SEGMENT,I5)        TB 133
          END                                                               TB 134-
```

TEST

PURPOSE

To compute the relative difference of two numerical integration results for the Romberg variable-interval-width integration routines.

METHOD

The first numerical integration result is the complex number (F1R, F1I) and the second is (F2R, F2I). The real and imaginary parts of the two results are subtracted and the differences are divided by the largest of F2R, F2I, DMIN or 10^{-37}. The denominator is chosen to avoid trying to maintain a small relative error for a quantity that is insignificantly small.

SYMBOL DICTIONARY

ABS	=	external routine (absolute value)
DEN	=	largest of $|F2R|$ and $|F2I|$
DMIN	=	minimum denominator
F1I	=	imaginary part of first integration result
F1R	=	real part of first integration result
F2I	=	imaginary part of second integration result
F2R	=	real part of second integration result
TI	=	relative difference of imaginary parts
TR	=	relative difference of real parts

CONSTANT

1.E-37 = tolerance in test for zero

TEST

```
      SUBROUTINE TEST (F1R,F2R,TR,F1I,F2I,TI,DMIN)
C
C     TEST FOR CONVERGENCE IN NUMERICAL INTEGRATION
C
      DEN=ABS(F2R)
      TR=ABS(F2I)
      IF (DEN.LT.TR) DEN=TR
      IF (DEN.LT.DMIN) DEN=DMIN
      IF (DEN.LT.1.E-37) GO TO 1
      TR=ABS((F1R-F2R)/DEN)
      TI=ABS((F1I-F2I)/DEN)
      RETURN
    1 TR=0.
      TI=0.
      RETURN
      END
```

TRIO

PURPOSE

To evaluate each of the parts of current expansion functions on a single segment due to each of the segments connected to the given segment.

METHOD

TRIO consists of a loop that uses the connection data in arrays ICON1 and ICON2 to locate all segments connected to segment J. Subroutine SBF is called to evaluate the current expansion function centered on each connected segment and on segment J. Only the function coefficients for that part of each expansion function on segment J are returned and are stored in arrays AX, BX, and CX. The number of the segment with which each expansion function part is associated is stored in array JCO and the total number of expansion functions involved is stored as JSNO.

SYMBOL DICTIONARY

IEND = -1 during calculations for end 1 of segment J, and +1 for end 2

JCOX = number of a segment connected to segment J

JEND = -1 if end 1 of segment JCOX is connected to segment J; +1 if end 2 of segment JCOX is connected to segment J

JMAX = dimension of the arrays in COMMON/SEGJ/

TRIO

```
      SUBROUTINE TRIO (J)                                               TR   1
C     COMPUTE THE COMPONENTS OF ALL BASIS FUNCTIONS ON SEGMENT J        TR   2
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300 TR   3
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX( TR   4
     2300),WLAM,IPSYM                                                   TR   5
      COMMON /SEGJ/ AX(30),BX(30),CX(30),JCO(30),JSNO,ISCON(50),NSCON,IP TR   6
     1CON(10),NPCON                                                     TR   7
      DATA JMAX/30/                                                     TR   8
      JSNO=0                                                            TR   9
      JCOX=ICON1(J)                                                     TR  10
      IF (JCOX.GT.10000) GO TO 7                                        TR  11
      JEND=-1                                                           TR  12
      IEND=-1                                                           TR  13
      IF (JCOX) 1,7,2                                                   TR  14
1     JCOX=-JCOX                                                        TR  15
      GO TO 3                                                           TR  16
2     JEND=-JEND                                                        TR  17
3     IF (JCOX.EQ.J) GO TO 6                                            TR  18
      JSNO=JSNO+1                                                       TR  19
      IF (JSNO.GE.JMAX) GO TO 9                                         TR  20
      CALL SBF (JCOX,J,AX(JSNO),BX(JSNO),CX(JSNO))                      TR  21
      JCO(JSNO)=JCOX                                                    TR  22
      IF (JEND.EQ.1) GO TO 4                                            TR  23
      JCOX=ICON1(JCOX)                                                  TR  24
      GO TO 5                                                           TR  25
4     JCOX=ICON2(JCOX)                                                  TR  26
5     IF (JCOX) 1,9,2                                                   TR  27
6     IF (IEND.EQ.1) GO TO 8                                            TR  28
7     JCOX=ICON2(J)                                                     TR  29
      IF (JCOX.GT.10000) GO TO 8                                        TR  30
      JEND=1                                                            TR  31
      IEND=1                                                            TR  32
      IF (JCOX) 1,8,2                                                   TR  33
8     JSNO=JSNO+1                                                       TR  34
      CALL SBF (J,J,AX(JSNO),BX(JSNO),CX(JSNO))                         TR  35
      JCO(JSNO)=J                                                       TR  36
      RETURN                                                            TR  37
9     PRINT 10, J                                                       TR  38
      STOP                                                              TR  39
C                                                                       TR  40
10    FORMAT (44H TRIO - SEGMENT CONNENTION ERROR FOR SEGMENT,I5)       TR  41
      END                                                               TR  42-
```

UNERE

PURPOSE

To calculate the electric field due to unit currents in the $\hat{t}_1$ and $\hat{t}_2$ directions on a surface patch.

METHOD

The electric field due to a patch j is calculated by the expression

$$\overline{E}(\overline{r}_0) = \frac{\eta_0}{i8\pi^2}\left[\left(\frac{-1 - i2\pi R/\lambda + 4\pi^2 (R/\lambda)^2}{(R/\lambda)^3}\right)\overline{J}_j + \left(\frac{3 + i6\pi R/\lambda - 4\pi^2 (R/\lambda)^2}{(R/\lambda)^5}\right)\overline{J}_j \cdot (\overline{R}/\lambda)(\overline{R}/\lambda)\right] \exp(-i2\pi R/\lambda)\frac{\Delta A_j}{\lambda^2},$$

where $i = \sqrt{-1}$, $\overline{J}_j = J_{1j}\hat{t}_{1j} + J_{2j}\hat{t}_{2j}$, $\overline{R}$ is the vector from the source to the observation point, and ΔA_j is the area of the patch. For UNERE, J_{1j} and J_{2j} are unity. The expression above for a single patch is obtained from the surface integral in equation (3) in Part I where constant current and one step integration are used for the patch.

CODING

UE14 - UE20 z components of patch parameters are adjusted for direct or reflected fields.

UE25 - UE32 For $R < 10^{-10}$, the fields are set to zero.

UE34 - UE47 Expression for $\overline{E}$ is evaluated for $\overline{J}_j$ equal to $\hat{t}_1$ and $\hat{t}_2$.

UE50 - UE55 For reflection in a perfect ground, $\overline{E}$ is reversed in sign.

UE57 - UE79 For reflection in an imperfect ground, $\overline{E}$ is multiplied by the reflection coefficients.

SYMBOL DICTIONARY

CONST = $\frac{\eta_0}{8\pi^2}$

CTH = $\cos\theta$; θ is the angle between the reflected ray and the normal to the surface

EDP = $(\overline{E} \cdot \hat{p})(R_H - R_V)$

UNERE

ER	=	$\frac{\eta_0}{i8\pi^2} \exp(-i\ 2\pi\ R/\lambda)\ \Delta A_j/\lambda^2$ at UE37
		= Q2 $(\hat{t}_{1j} \cdot \bar{R}/\lambda)$ at UE40
		= Q2 $(\hat{t}_{2j} \cdot \bar{R}/\lambda)$ at UE44
EXK, EYK, EZK	=	$\bar{E}$ due to current $\hat{t}_{1j}$
EXS, EYS, EZS	=	$\bar{E}$ due to current $\hat{t}_{2j}$
IPGND	=	flag to cause computation of reflected field when equal to 2
PX, PY	=	$\hat{p}$; unit vector normal to the plane of incidence of the reflected ray
Q1	=	$\left[\frac{-1 - i2\pi R/\lambda + 4\pi^2(R/\lambda)^2}{(R/\lambda)^3}\right]$ (ER)
Q2	=	$\left[\frac{3 + i6\pi R/\lambda - 4\pi^2(R/\lambda)^2}{(R/\lambda)^5}\right]$ (ER)
R	=	R/λ
RRH	=	R_H
RRV	=	R_V
RT	=	$(R/\lambda)^3$
RX, RY, RZ	=	$\bar{R}/\lambda$
R2	=	$(R/\lambda)^2$
S	=	$\Delta A_j/\lambda^2$
T1XJ, T1YJ, T1ZJ	=	$\hat{t}_{1j}$
T2XJ, T2YJ, T2ZJ	=	$\hat{t}_{2j}$
TPI	=	2π
TT1	=	$-2\pi R/\lambda$
TT2	=	$4\pi 2(R/\lambda)^2$

$$\left.\begin{array}{l}\text{XOB}\\ \text{YOB}\\ \text{ZOB}\end{array}\right\} = \text{field evaluation point}$$

XYMAG = magnitude of the projection of $\overline{R}/\lambda$ onto the x-y plane

ZR = z component of $\overline{R}/\lambda$ after reflection

CONSTANTS

$4.771341188 = \frac{\eta_0}{8\pi^2}$

$6.283185308 = 2\pi$

```
      SUBROUTINE UNERE (XOB,YOB,ZOB)
C     CALCULATES THE ELECTRIC FIELD DUE TO UNIT CURRENT IN THE T1 AND T2
C     DIRECTIONS ON A PATCH
      COMPLEX EXK,EYK,EZK,EXS,EYS,EZS,EXC,EYC,EZC,ZRATI,ZRATI2,T1,ER,Q1,
     1Q2,RRV,RRH,EDP,FRATI
      COMMON /DATAJ/ S,B,XJ,YJ,ZJ,CABJ,SABJ,SALPJ,EXK,EYK,EZK,EXS,EYS,EZ
     1S,EXC,EYC,EZC,RKH,IEXK,IND1,IND2,IPGND
      COMMON /GND/ZRATI,ZRATI2,FRATI,CL,CH,SCRWL,SCRWR,NRADL,KSYMP,IFAR,
     1IPERF,T1,T2
      EQUIVALENCE (T1XJ,CABJ), (T1YJ,SABJ), (T1ZJ,SALPJ), (T2XJ,B), (T2Y
     1J,IND1), (T2ZJ,IND2)
      DATA TPI,CONST/6.283185308,4.771341188/
C     CONST=ETA/(8.*PI**2)
      ZR=ZJ
      T1ZR=T1ZJ
      T2ZR=T2ZJ
      IF (IPGND.NE.2) GO TO 1
      ZR=-ZR
      T1ZR=-T1ZR
      T2ZR=-T2ZR
    1 RX=XOB-XJ
      RY=YOB-YJ
      RZ=ZOB-ZR
      R2=RX*RX+RY*RY+RZ*RZ
      IF (R2.GT.1.E-20) GO TO 2
      EXK=(0.,0.)
      EYK=(0.,0.)
      EZK=(0.,0.)
      EXS=(0.,0.)
      EYS=(0.,0.)
      EZS=(0.,0.)
      RETURN
    2 R=SQRT(R2)
      TT1=-TPI*R
      TT2=TT1*TT1
      RT=R2*R
      ER=CMPLX(SIN(TT1),-COS(TT1))*(CONST*S)
      Q1=CMPLX(TT2-1.,TT1)*ER/RT
      Q2=CMPLX(3.-TT2,-3.*TT1)*ER/(RT*R2)
      ER=Q2*(T1XJ*RX+T1YJ*RY+T1ZR*RZ)
      EXK=Q1*T1XJ+ER*RX
      EYK=Q1*T1YJ+ER*RY
      EZK=Q1*T1ZR+ER*RZ
      ER=Q2*(T2XJ*RX+T2YJ*RY+T2ZR*RZ)
      EXS=Q1*T2XJ+ER*RX
      EYS=Q1*T2YJ+ER*RY
      EZS=Q1*T2ZR+ER*RZ
      IF (IPGND.EQ.1) GO TO 6
      IF (IPERF.NE.1) GO TO 3
      EXK=-EXK
      EYK=-EYK
      EZK=-EZK
      EXS=-EXS
      EYS=-EYS
      EZS=-EZS
      GO TO 6
    3 XYMAG=SQRT(RX*RX+RY*RY)
      IF (XYMAG.GT.1.E-6) GO TO 4
      PX=0.
      PY=0.
      CTH=1.
      RRV=(1.,0.)
      GO TO 5
    4 PX=-RY/XYMAG
```

```
        PY=RX/XYMAG                                          UN  65
        CTH=RZ/SQRT(XYMAG*XYMAG+RZ*RZ)                       UN  66
        RRV=CSQRT(1.-ZRATI*ZRATI*(1.-CTH*CTH))               UN  67
    5   RRH=ZRATI*CTH                                        UN  68
        RRH=(RRH-RRV)/(RRH+RRV)                              UN  69
        RRV=ZRATI*RRV                                        UN  70
        RRV=-(CTH-RRV)/(CTH+RRV)                             UN  71
        EDP=(EXK*PX+EYK*PY)*(RRH-RRV)                        UN  72
        EXK=EXK*RRV+EDP*PX                                   UN  73
        EYK=EYK*RRV+EDP*PY                                   UN  74
        EZK=EZK*RRV                                          UN  75
        EDP=(EXS*PX+EYS*PY)*(RRH-RRV)                        UN  76
        EXS=EXS*RRV+EDP*PX                                   UN  77
        EYS=EYS*RRV+EDP*PY                                   UN  78
        EZS=EZS*RRV                                          UN  79
    6   RETURN                                               UN  80
        END                                                  UN  81-
```

WIRE

PURPOSE

To compute segment coordinates to fill COMMON/DATA/ for a straight line of segments.

METHOD

The formal parameters specify the beginning and ending points of the line and the number of segments into which it is to be divided. The code computes the coordinates of the end points of each segment. The lengths of successive segments are scaled by the factor RDEL if this factor is not one. For NS segments, the length of the first segment is

$$S_1 = \frac{L(1 - RDEL)}{1 - (RDEL)^{NS}}$$

or $$S_1 = L/NS \text{ if } RDEL = 1$$

where L is the total length of wire.

The radius is RAD for the first segment and is scaled by RRAD.

SYMBOL DICTIONARY

DELZ	=	segment length
FNS	=	real number equivalent of NS
IST	=	initial segment number
ITG	=	tag number assigned to all segments of the line
NS	=	number of segments into which line is divided
RAD	=	radius of first segment
RADZ	=	segment radius
RD, RDEL	=	scaling factor for segment length
RRAD	=	scaling factor for segment radius
XD	=	increment to x coordinates
XS1	=	x coordinate of first end of segment
XS2	=	x coordinate of second end of segment
XW1	=	x coordinate of first end of line
XW2	=	x coordinate of second end of line

X2(I)	= x coordinate of end 2 of segment I
YD	= increment to y coordinates
YS1	= y coordinate of first end of segment
YS2	= y coordinate of second end of segment
YW1	= y coordinate of first end of wire
YW2	= y coordinate of second end of wire
Y2(I)	= y coordinate of end 2 of segment I
ZD	= increment to z coordinates
ZS1	= z coordinate of first end of segment
ZS2	= z coordinate of second end of segment
ZW1	= z coordinate of first end of line
ZW2	= z coordinate of second end of line
Z2(I)	= z coordinate of second end of segment I

```
      SUBROUTINE WIRE (XW1,YW1,ZW1,XW2,YW2,ZW2,RAD,RDEL,RRAD,NS,ITG)
C
C     SUBROUTINE WIRE GENERATES SEGMENT GEOMETRY DATA FOR A STRAIGHT
C     WIRE OF NS SEGMENTS.
C
      COMMON /DATA/ LD,N1,N2,N,NP,M1,M2,M,MP,X(300),Y(300),Z(300),SI(300
     1),BI(300),ALP(300),BET(300),ICON1(300),ICON2(300),ITAG(300),ICONX(
     2300),WLAM,IPSYM
      DIMENSION X2(1), Y2(1), Z2(1)
      EQUIVALENCE (X2(1),SI(1)), (Y2(1),ALP(1)), (Z2(1),BET(1))
      IST=N+1
      N=N+NS
      NP=N
      MP=M
      IPSYM=0
      IF (NS.LT.1) RETURN
      XD=XW2-XW1
      YD=YW2-YW1
      ZD=ZW2-ZW1
      IF (ABS(RDEL-1.).LT.1.E-6) GO TO 1
      DELZ=SQRT(XD*XD+YD*YD+ZD*ZD)
      XD=XD/DELZ
      YD=YD/DELZ
      ZD=ZD/DELZ
      DELZ=DELZ*(1.-RDEL)/(1.-RDEL**NS)
      RD=RDEL
      GO TO 2
1     FNS=NS
      XD=XD/FNS
      YD=YD/FNS
      ZD=ZD/FNS
      DELZ=1.
      RD=1.
2     RADZ=RAD
      XS1=XW1
      YS1=YW1
      ZS1=ZW1
      DO 3 I=IST,N
      ITAG(I)=ITG
      XS2=XS1+XD*DELZ
      YS2=YS1+YD*DELZ
      ZS2=ZS1+ZD*DELZ
      X(I)=XS1
      Y(I)=YS1
      Z(I)=ZS1
      X2(I)=XS2
      Y2(I)=YS2
      Z2(I)=ZS2
      BI(I)=RADZ
      DELZ=DELZ*RD
      RADZ=RADZ*RRAD
      XS1=XS2
      YS1=YS2
3     ZS1=ZS2
      X2(N)=XW2
      Y2(N)=YW2
      Z2(N)=ZW2
      RETURN
      END
```

ZINT

PURPOSE

To compute the internal impedance of a circular wire with finite conductivity.

METHOD

The internal impedance per unit length of a circular wire is given by

$$Z = \frac{j}{b}\sqrt{\frac{f\mu}{2\pi\sigma}}\left[\frac{Ber(q) + jBei(q)}{Ber'(q) + jBei'(q)}\right],$$

where

$q = b\sqrt{2\pi f\mu\sigma}$

σ = wire conductivity

μ = permeability of free space

b = wire radius

f = frequency

$\left.\begin{matrix} Ber \\ Bei \end{matrix}\right\}$ = Kelvin functions

The term that modifies the diagonal matrix element G_{ii} in the interaction matrix is the total impedance of segment i divided by Δ_i/λ, where Δ_i = segment length. Thus, if G_{ii} is the diagonal matrix element without loading, the new element is

$$G_{ii} - Z\Delta_i/(\Delta_i/\lambda) = G_{ii} - Z\lambda .$$

Normalized to wavelength, this term is

$$Z_i = Z\lambda = \frac{j}{(b/\lambda)}\sqrt{\frac{c\mu}{2\pi(\sigma\lambda)}}\left[\frac{Ber(q) + jBei(q)}{Ber'(q) + jBei'(q)}\right],$$

where

$q = (b/\lambda)\sqrt{2\pi c\mu(\sigma\lambda)}$

c = velocity of light

The Kelvin functions and derivatives of Kelvin functions are computed from their polynomial approximations.

CODING

ZI8 - ZI15 Functions θ, ϕ, f, and g for large argument polynomial approximations (see ref. 5).

ZI19 - ZI26 Compute Ber(q) + jBei(q) for $q \leq 8$.

ZI27 - ZI31 Compute Ber'(q) + jBei'(q) for $q \leq 8$.

ZI32 [Ber(q) + jBei(q)]/[Ber'(q) + jBei'(q)].

ZI34 Ber(q) + jBei(q) for $8 < q \leq 110$.

ZI35 Ber'(q) + jBei'(q) for $8 < q \leq 110$.

ZI36 [Ber(q) + jBei(q)]/[Ber'(q) + jBei'(q)].

ZI38 [Ber(q) + jBei(q)]/[Ber'(q) + jBei'(q)] for $110 < q < \infty$.

ZI39 Computation of Z_i.

SYMBOL DICTIONARY

BEI = Bei(q) or Bei'(q)
BER = Ber(q) or Ber'(q)
BR1 = Ber(q) + jBei(q) or [Ber(q) + jBei(q)]/[Ber'(q) + Bei'(q)]
BR2 = Ber'(q) + jBei'(q)
CEXP = external routine [exp(complex argument)]
CMOTP = $c\mu/(2\pi)$
CMPLX = external routine (forms complex number)
CN = $(1 + j)/\sqrt{2}$
D = function argument
F(D) = f(D) (see ref. 5)
FJ = j
G(D) = g(D) (see ref. 5)
PH(D) = $\phi(X)$, D = 8/X (see ref. 5)
PI = π
POT = $\pi/2$
ROLAM = b/λ
S = $(X/8)^4$
SIGL = $\sigma\lambda$
SQRT = external routine (square root)
TH(D) = $\theta(X)$, D = 8/X (see ref. 5)
TP = 2π

TPCMU = $2\pi c\mu$; c = velocity of light

X = q

Y = $(X/8)^2$

ZINT = Z_i

CONSTANTS

1.5707963	= $\pi/2$
3.141592654	= π
6.283185308	= 2π
60.	= $c\mu/2\pi$
2.368705E+3	= $2\pi c\mu$
(0., 1.)	= j
(0.70710678, 0.70710678)	= $(1 + j)/\sqrt{2}$
(0.70710678, -0.70710678)	= limit for $q \to \infty$ of [Ber(q) + jBei(q)]/ [Ber'(q) + jBei'(q)]

Other constants are factors in the polynomial approximations.

ZINT

```
      COMPLEX FUNCTION ZINT(SIGL,ROLAM)                                 ZI   1
C                                                                       ZI   2
C     ZINT COMPUTES THE INTERNAL IMPEDANCE OF A CIRCULAR WIRE           ZI   3
C                                                                       ZI   4
C                                                                       ZI   5
      COMPLEX TH,PH,F,G,FJ,CN,BR1,BR2                                   ZI   6
      COMPLEX CC1,CC2,CC3,CC4,CC5,CC6,CC7,CC8,CC9,CC10,CC11,CC12,CC13,CC ZI   7
     114                                                                ZI   8
      DIMENSION FJX(2), CNX(2), CCN(28)                                 ZI   9
      EQUIVALENCE (FJ,FJX), (CN,CNX), (CC1,CCN(1)), (CC2,CCN(3)), (CC3,C ZI  10
     1CN(5)), (CC4,CCN(7)), (CC5,CCN(9)), (CC6,CCN(11)), (CC7,CCN(13)),  ZI  11
     2(CC8,CCN(15)), (CC9,CCN(17)), (CC10,CCN(19)), (CC11,CCN(21)), (CC1 ZI  12
     32,CCN(23)), (CC13,CCN(25)), (CC14,CCN(27))                        ZI  13
      DATA PI,POT,TP,TPCMU/3.1415926,1.5707963,6.2831853,2.368705E+3/   ZI  14
      DATA CMOTP/60.00/,FJX/0.,1./,CNX/.70710678,.70710678/             ZI  15
      DATA CCN/6.E-7,1.9E-6,-3.4E-6,5.1E-6,-2.52E-5,0.,-9.06E-5,-9.01E-5 ZI  16
     1,0.,-9.765E-4,.0110486,-.0110485,0.,-.3926991,1.6E-6,-3.2E-6,1.17E ZI  17
     2-5,-2.4E-6,3.46E-5,3.38E-5,5.E-7,2.452E-4,-1.3813E-3,1.3811E-3,-6. ZI  18
     3325001E-2,-1.E-7,.7071068,.7071068/                               ZI  19
      TH(D)=(((((CC1*D+CC2)*D+CC3)*D+CC4)*D+CC5)*D+CC6)*D+CC7           ZI  20
      PH(D)=(((((CC8*D+CC9)*D+CC10)*D+CC11)*D+CC12)*D+CC13)*D+CC14      ZI  21
      F(D)=SQRT(POT/D)*CEXP(-CN*D+TH(-8./X))                            ZI  22
      G(D)=CEXP(CN*D+TH(8./X))/SQRT(TP*D)                               ZI  23
      X=SQRT(TPCMU*SIGL)*ROLAM                                          ZI  24
      IF (X.GT.110.) GO TO 2                                            ZI  25
      IF (X.GT.8.) GO TO 1                                              ZI  26
      Y=X/8.                                                            ZI  27
      Y=Y*Y                                                             ZI  28
      S=Y*Y                                                             ZI  29
      BER=((((((-9.01E-6*S+1.22552E-3)*S-.08349609)*S+2.6419140)*S-32.36 ZI  30
     113456)*S+113.77778)*S-64.)*S+1.                                   ZI  31
      BEI=((((((1.1346E-4*S-.01103667)*S+.52185615)*S-10.567658)*S+72.81 ZI  32
     17777)*S-113.77778)*S+16.)*Y                                       ZI  33
      BR1=CMPLX(BER,BEI)                                                ZI  34
      BER=(((((((-3.94E-6*S+4.5957E-4)*S-.02609253)*S+.66047849)*S-6.068 ZI  35
     111481)*S+14.222222)*S-4.)*Y)*X                                    ZI  36
      BEI=((((((4.609E-5*S-3.79386E-3)*S+.14677204)*S-2.3116751)*S+11.37 ZI  37
     117778)*S-10.666667)*S+.5)*X                                       ZI  38
      BR2=CMPLX(BER,BEI)                                                ZI  39
      BR1=BR1/BR2                                                       ZI  40
      GO TO 3                                                           ZI  41
    1 BR2=FJ*F(X)/PI                                                    ZI  42
      BR1=G(X)+BR2                                                      ZI  43
      BR2=G(X)*PH(8./X)-BR2*PH(-8./X)                                   ZI  44
      BR1=BR1/BR2                                                       ZI  45
      GO TO 3                                                           ZI  46
    2 BR1=CMPLX(.70710678,-.70710678)                                   ZI  47
    3 ZINT=FJ*SQRT(CMOTP/SIGL)*BR1/ROLAM                                ZI  48
      RETURN                                                            ZI  49
      END                                                               ZI  50-
```

Section III
Common Blocks

This section discusses each labeled common block which is used in the NEC-2 code. For each common block, a list of the routines in which it is used is given along with a definition of the variables used in conjunction with the common block. The common blocks are presented in alphabetical order.

COMMON/ANGL/ SALP(300)

Routines Using /ANGL/

CABC, CMSS, CMSW, CMWS, CMWW, DATAGN, ETMNS, FFLD, GFIL, GFLD, GFOUT, MOVE, NEFLD, NHFLD, PATCH, QDSRC, REFLC

/ANGL/ Parameters for Wire Segments

SALP(I) = sin (α), where α = elevation angle of segment I (see figure 11)

/ANGL/ Parameters for Surface Patches

SALP(LD - I + 1) = +1 if $\hat{t}_1 \times \hat{t}_2 = \hat{n}$ for patch I, or

-1 if $\hat{t}_1 \times \hat{t}_2 = -\hat{n}$ for patch I

The second case occurs when the patch has been produced by reflection of a patch originally input.

COMMON/CMB/ CM(4000)

Routines Using /CMB/

MAIN, GFIL, GFOUT

The interaction matrix is stored in array CM. If the matrix is too large to fit in CM, then pairs of blocks of the matrix are stored in CM as they are needed.

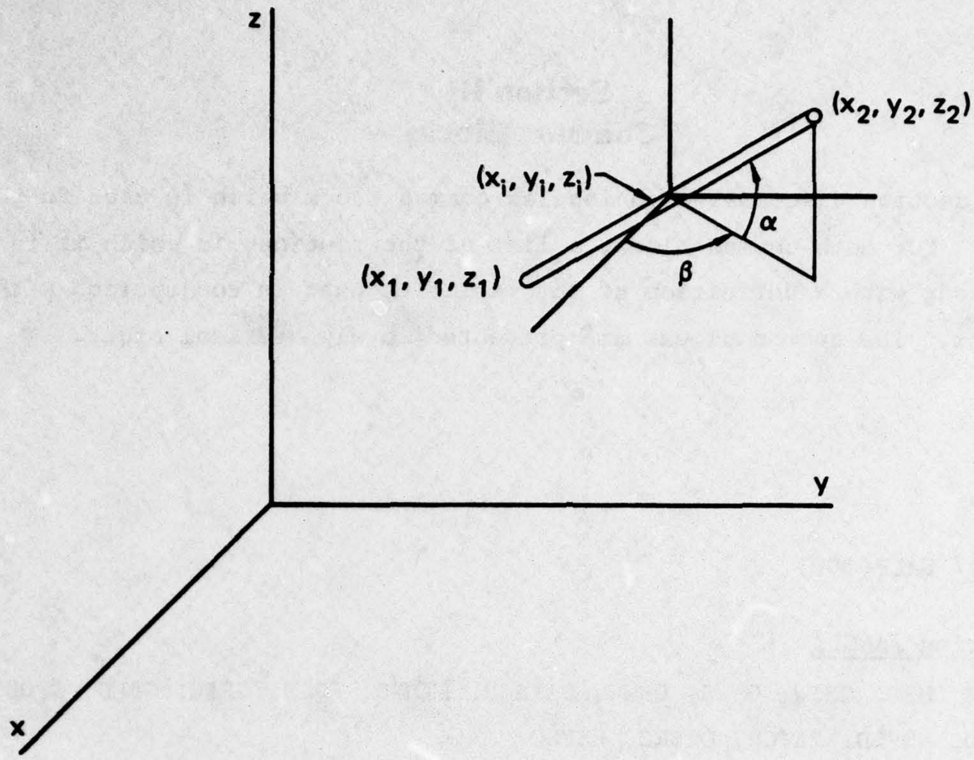


Figure 11. Coordinates of Segment i.

COMMON/CRNT/AIR(300), AII(300), BIR(300), BII(300), CIR(300), CII(300)
CUR(900)

Routines Using /CRNT/

MAIN, CABC, FFLD, GFLD, NEFLD, NETWK, NHFLD

/CRNT/ Parameters for Wire Segments

Subroutine CABC fills the first six arrays in /CRNT/ with the real and imaginary parts of the constants in the current expansion of each segment,

$$I_i(s) = A_i + B_i \sin [k(s - s_i)] + C_i \cos [k(s - s_i)] ,$$

where $s = s_i$ at the center of segment i. Except during intermediate calculations for non-radiating networks, the current basis-function amplitudes are computed and stored in array CUR. CABC replaces the basis function amplitudes in CUR by the current at the center of each segment, $(A_i + C_i)$. For i = I,

$\left.\begin{array}{l}\text{AIR(I)}\\ \text{AII(I)}\end{array}\right\}$ = A_i/λ (real, imaginary)

$\left.\begin{array}{l}\text{BIR(I)}\\ \text{BII(I)}\end{array}\right\}$ = B_i/λ

$\left.\begin{array}{l}\text{CIR(I)}\\ \text{CII(I)}\end{array}\right\}$ = C_i/λ

CUR(I) = amplitude of i^{th} basis function going into CABC or $(A_i + C_i)/\lambda$ at end of CABC

/CRNT/ Parameters for Surface Patches

Surface current components are stored in CUR. Before CABC is called, the surface current strengths in directions $\hat{t}_1$ and $\hat{t}_2$ on patch i are stored in CUR(N + 2I - 1) and CUR(N + 2I), respectively where N is the number of segments. After CABC, the x, y and z components of surface current are stored in CUR(N + 3I - 2), CUR(N + 3I - 1) and CUR(N + 3I), respectively.

COMMON/DATA/ LD, N1, N2, N, NP, M1, M2, M, MP, X(300), Y(300), Z(300), SI(300), BI(300), ALP(300), BET(300), ICON1(300), ICON2(300), ITAG(300), ICONX(300), WLAM, IPSYM

Routines Using /DATA/

MAIN, ARC, CABC, CMNGF, CMSET, CMSS, CMSW, CMWS, CMWW, CONECT, DATAGN, ETMNS, FFLD, FFLDS, GFIL, GFLD, GFOUT, ISEGNO, LOAD, MOVE, NEFLD, NETWK, NFPAT, NHFLD, PATCH, QDSRC, RDPAT, REFLC, SBF, TBF, TRIO, WIRE

/DATA/ Parameters for Wire Segments

The arrays in /DATA/ are used to store the parameters defining the segments. Two forms of the segment parameters are used.

During geometry input in routines ARC, CONECT, DATAGN, MOVE, REFLEC and WIRE, the coordinates of the segment ends are stored. The symbol meanings in the geometry routines are:

X(I) = X_1
Y(I) = Y_1
Z(I) = Z_1

SI(I) = X_2 [equivalenced to X2(I)]

ALP(I) = Y_2 [equivalenced to Y2(I)]

BET(I) = Z_2 [equivalenced to Z2(I)]

where X_1, Y_1, Z_1 are the coordinates of the first end of the segment, and X_2, Y_2, Z_2 are the coordinates of the second end, as illustrated in figure 11. Coordinates may have any units but must be scaled to meters before data input is ended, since the main program requires meters.

In the main program, the segment data is converted to: the coordinates of the segment center, components of the unit vector in the direction of the segment, and the segment length. The symbol meanings after the geometry section are:

$$\left.\begin{array}{l} \text{X(I)} \\ \text{Y(I)} \\ \text{Z(I)} \end{array}\right\} = X_i,\ Y_i,\ Z_i \text{ (see figure 11.)}$$

SI(I) = segment length

ALP(I) = $\cos\alpha \cos\beta$ [equivalenced to CAB(I)]

BET(I) = $\cos\alpha \sin\beta$ [equivalenced to SAB(I)]

The z component of the unit vector in the direction of the segment, $\sin\alpha$, is stored in /ANGL/.

The other symbol meanings in /DATA/ for segments are:

BI(I) = radius of segment I

ICON1(I) = connection number for end 1 of segment I. If k is a positive integer less than 10,000, the meaning of ICON1 is as follows.

ICON1(I) = 0: no connection.

ICON1(I) = ±k: end 1 connects to segment k. If more than one segment connects to end 1 of segment I, then k is the number of the next connected segment encountered by starting at I and going through the list of segments in cyclic order. ICON1(I) = +k: parallel reference directions with end 2 of the other segment connecting to end 1 of segment I. ICON1(I) = -k: opposed reference directions.

ICON1(I) = I: end 1 of segment I connects to a ground plane.

ICON1(I) = 10,000 +k: end 1 of segment I connects to a surface with the 4 patches around the connection point numbered k, k + 1, k + 2 and k + 3.

ICON2(I) = connection number for end 2 of segment I.

ITAG(I) = tag number of segment I. This number is assigned during structure input to permit later reference to the segment without knowing the segment index I in the data arrays.

ICONX(I) = equation number for the new basis function when segment I is in a numerical Green's function file and a new segment connects to segment I modifying the old basis function.

/DATA/ Parameters for Surface Patches

Patch parameters are set in subroutine PATCH. The input parameters for a patch are the coordinates of the patch center, patch area, and orientation of the outward, normal unit vector, $\hat{n}$. The parameters stored in /DATA/ are the center point coordinates, area, and the components of the two surface unit vectors, $\hat{t}_1$ and $\hat{t}_2$. The vector $\hat{t}_1$ is parallel to a side of the triangular, rectangular, or quadrilateral patch. For a patch of arbitrary shape, it is chosen by the following rules:

For a horizontal patch, $\hat{t}_1 = \hat{x}$;

For a nonhorizontal patch, $\hat{t}_1 = (\hat{z} \times \hat{n})/|\hat{z} \times \hat{n}|$;

$\hat{t}_2$ is then chosen as $\hat{t}_2 = \hat{n} \times \hat{t}_1$

With J = LD + 1 - I, the parameters for patch I are stored as follows.

X(J), Y(J), Z(J) = x, y, and z coordinates of the patch center

SI(J), ALP(J), BET(J) = x, y, z components of $\hat{t}_1$ (equivalenced to T1X, T1Y, T1Z)

ICON1(J), ICON2(J), ITAG(J) = x, y, and z components of $\hat{t}_2$ (equivalenced to T2X, T2Y, T2Z)

BI(J) = patch area

Scalar variables in /DATA/ are:

IPSYM = symmetry flag. The meanings of IPSYM are:

IPSYM = 0: no symmetry

IPSYM > 0: plane symmetry

IPSYM < 0: cylindrical symmetry

IPSYM = 2: plane symmetry about Z = 0

|IPSYM| > 2: structure has been rotated about x or y axis. If ground plane is indicated by IGND ≠ 0 in the call to subroutine CONECT and IPSYM = 2, symmetry about a horizontal plane is removed by multiplying NP by 2. If |IPSYM| > 2 and IGND ≠ 0, all symmetry is removed by setting NP = N and IPSYM = 0 in CONECT.

LD = length of arrays in /DATA/

N1 = number of segments in NGF. If NGF is not used N1 = 0

N2 = N1 + 1

N = total number of segments

NP = number of segments in a symmetric cell

M1 = number of patches in NGF. If NGF is not used M1 = 0

M2 = M1 + 1

M = total number of patches

MP = number of patches in a symmetric cell

WLAM = wavelength in meters

COMMON/DATAJ/ S, B, XJ, YJ, ZJ, CABJ, SABJ, SALPJ, EXK, EYK, EZK, EXS, EYS, EZS, EXC, EYC, EZC, RKH, IEXK, IND1, IND2, IPGND

Routines Using /DATAJ/

CMNGF, CMSET, CMSS, CMSW, CMWS, CMWW, EFLD, HINTG, HSFLD, NEFLD, NHFLD, PCINT, QDSRC, SFLDS, UNERE

/DATAJ/ is used to pass the parameters of the source segment or patch to the routines that compute the E or H field and to return the field components.

/DATAJ/ Parameters for Wire Segments

S = segment length

B = segment radius

XJ, YJ, ZJ = coordinates of segment center

CABJ, SABJ, SALPJ = x, y, and z, respectively, of the unit vector in the direction of the segment

EXK, EYK, EZK = x, y, and z components of the E or H field due to a constant current

EXS, EYS, EZS = x, y, and z components of the E or H field due to a sin ks current

EXC, EYC, EZC = x, y, and z components of the E or H field due to cos ks current

RKH = minimum distance for use of the Hertzian dipole approximation for computing the E field of a segment

IEXK = flag to select thin wire approximation or extended thin wire approximation for E field (IEXK = 1 for extended thin wire approximation)

IND1 = flag to inhibit use of the extended thin wire approximation on end 1 of the source segment. This is used when there is a bend or change in radius at end 1. IND1 = 2 inhibits the extended thin wire approximation.

IND2 = flag to inhibit use of the extended thin wire approximation on end 2 of the source segment

IPGND = not used

/DATAJ/ Parameters for Surface Patches

S = patch area in units of wavelength squared

B = x component of $\hat{t}_2$ for the patch

XJ, YJ, ZJ = x, y, and z components of the position of the patch center

$\left.\begin{array}{l}\text{CABJ}\\ \text{SABJ}\\ \text{SALPJ}\end{array}\right\}$ = x, y, and z components of $\hat{t}_1$

$\left.\begin{array}{l}\text{EXK}\\ \text{EYK}\\ \text{EZK}\end{array}\right\}$ = x, y, and z components of $\bar{E}$ or $\bar{H}$ due to a current with unit magnitude in the direction $\hat{t}_1$ on the patch

$\left.\begin{array}{l}\text{EXS}\\ \text{EYS}\\ \text{EZS}\end{array}\right\}$ = $\bar{E}$ or $\bar{H}$ due to a current $\hat{t}_2$ on the patch

$\left.\begin{array}{l}\text{EXC}\\ \text{EYC}\\ \text{EZC}\end{array}\right\}$ = not used; may serve as intermediate variables in some routines

IND1 = y component of $\hat{t}_2$

IND2 = z component of $\hat{t}_2$

IPGND = flag to request calculation of the direct field or field reflected from the ground (two for ground)

COMMON/FPAT/ NTH, NPH, IPD, IAVP, INOR, IAX, THETS, PHIS, DTH, DPH, RFLD, GNOR, CLT, CHT, EPSR2, SIG2, IXTYP, XPR6, PINR, PNLR, PLOSS NEAR, NFEH, NRX, NRY, NRZ, XNR, YNR, ZNR, DXNR, DYNR, DZNR

Routines Using /FPAT/

MAIN, NFPAT, RDPAT

Variables are defined in subroutine descriptions.

COMMON/GGRID/ AR1(11, 10, 4), AR2(17, 5, 4), AR3(9, 8, 4), EPSCF, DXA(3), DYA(3), XSA(3), YSA(3), NXA(3), NYA(3)

Routines Using /GGRID/

MAIN, GFIL, GFOUT, INTRP

Variables are defined under subroutine INTRP.

COMMON/GND/ ZRATI, ZRATI2, FRATI, CL, CH, SCRWL, SCRWR, NRADL, KSYMP, IFAR, IPERF, T1, T2

Routines Using /GND/

MAIN, CMSW, EFLD, ETMNS, FFLD, GFIL, GFOUT, HINTG, HSFLD, NEFLD, RDPAT, SFLDS, UNERE

/GND/ contains parameters of the ground including the two-medium ground and radial-wire ground-screen cases. The symbol definitions are as follows.

ZRATI = $[\epsilon_r - j\sigma/\omega\epsilon_0]^{-1/2}$

where σ is ground conductivity (mhos/meter), ϵ_r is the relative dielectric constant, ϵ_0 is the permittivity of free space (farads/meter), and $\omega = 2\pi f$.

ZRATI2 = same as ZRATI, but for a second ground medium

FRATI = $(k_1^2 - k_2^2)/(k_1^2 + k_2^2)$ where $k_2 = \omega \sqrt{\mu_0 \epsilon_0}$ and $k_1 = k_2$/ZRATI

CL = distance in wavelengths of cliff edge from origin

CH = cliff height in wavelengths

SCRAWL = length of wires in radial-wire ground screen (normalized to wavelength)

SCRWR = radius of wires in screen in wavelengths

NRADL = number of radials in ground screen; zero implies no screen (input quantity, GN card)

KSYMP = ground flag (=1, no ground; =2, ground present)

IFAR = input integer flag on RP card; specifies type of field computation or type of ground system for far fields

IPERF = flag to select type of ground (see GN card)

T1, T2 = constants for the radial-wire ground-screen impedance

COMMON/GWAV/ U, U2, XX1, XX2, R1, R2, ZMH, ZPH

Routines Using /GWAV/

MAIN, GFLD, GWAVE, SFLDS

Symbol Definitions:

U = $(\epsilon_r - j\sigma/\omega\epsilon_0)^{-1/2}$

ϵ_r = relative dielectric constant; σ = conductivity of ground

U2 = U^2

XX1, XX2 : defined in GFLD and SFLDS

R1 = distance from current element to point at which field is evaluated

R2 = distance from image of current element to point at which field is evaluated

ZMH = Z - Z'

ZPH = Z + Z' where Z is height of the field evaluation point and Z' is the height of the current element

COMMON/INCOM/ XO, YO, ZO, SN XSN, YSN, ISNOR

Routines Using /INCOM/

EFLD, SFLDS

Symbol Definitions:

XO, YO, ZO = point at which field due to ground will be evaluated

SN = cos α (see Figure 11)

XSN = cos ß

YSN = sin ß

ISNOR = 1 to evaluate field due to ground by interpolation
0 to use Norton's approximation

COMMON/MATPAR/ ICASE, NBLOKS, NPBLK, NLAST, NBLSYM, NPSYM, NLSYM, IMAT, ICASX, NBBX, NPBX, NLBX, NBBL NPBL, NLBL

Routines Using /MATPAR/

MAIN, CMNGF, CMSET, FACGF, FACIO, FACTRS, FBLOCK, FBNGF, GFIL, GFOUT, LFACTR, LTSOLV, LUNSCR, REBLK, SOLGF, SOLVES

/MATPAR/ contains matrix blocking parameters for cases requiring file storage of the matrix. Symbol definitions in /MATPAR/ are as follows.

ICASE = storage mode for primary matrix, defined as follows.
1 unsymmetric matrix fits in core
2 symmetric matrix fits in core
3 unsymmetric matrix out of core
4 symmetric matrix out of core, but submatrices fit in core
5 symmetric matrix out of core, submatrices also out of core

NBLOKS = number of blocks of columns of the computed matrix (in core matrix, NBLOKS = 1)

NPBLK = number of columns in the first (NBLOKS - 1) blocks

NLAST = number of columns in the last block

NBLSYM, NPSYM, NLSYM = same function as the preceding three variables; however, in this case the parameters refer to the submatrix in the symmetry case

IMAT = storage reserved in CM for the primary NGF matrix A or a block of A (number of complex numbers)

ICASX = storage mode for NGF solution (see Section VII)

NBBX = number of blocks in matrix B stored by blocks of rows

NPBX = number of rows in a block of B stored by rows

NLBX = number of rows in the last block of B

NBBL = number of blocks in matrix C stored by rows (and number of blocks in B stored by columns)

NPBL = number of rows (columns) in a block of C (B)

NLBL = number of rows (columns) in the last block of C (B)

COMMON/NETCX/ ZPED, PIN, PNLS, NEQ, NPEQ, NEQ2, NONET, NTSOL, NPRINT, MASYM, ISEG1(30), ISEG2(30), X11R(30), X11I(30), X12R(30), X12I(30), X22R(30), X22I(30), NTYP(30)

Routines Using /NETCX/

MAIN, NETWK

Variables are defined under subroutine NETWK.

COMMON/SAVE/ IP(600), KCOM, COM(13,5), EPSR, SIG, SCRWLT, SCRWRT, FMHZ

Routines Using /SAVE/

MAIN, GFIL, GFOUT, RDPAT

Symbol Definitions:

IP = vector of indices of pivot elements used in factoring the matrix

KCOM = number of CM or CE data cards (maximum 5)

COM = array storing the contents of CM or CE cards

EPSR = relative dielectric constant of the ground

SIG = conductivity of the ground

SCRWLT = length of radials in radial wire ground screen approximation (meters)

SCRWRT = radius of wires in radial wire ground screen approximation (meters)

FMHZ = frequency in MHz

COMMON/SCRATM/D(600)

in routines CMSET, FACTR, LFACTR

COMMON/SCRATM/Y(600)

in routines LTSOLV, SOLGF, SOLVE, SOLVES

COMMON/SCRATM/GAIN(1200)

in routine RDPAT

Symbol Definitions:

D and Y =

complex vectors used in matrix decomposition and solution

GAIN = array to store antenna gain for subsequent normalization

COMMON/SEGJ/ AX(30), BX(30), CX(30), JCO(30), JSNO, ISCON(50), NSCON, IPCON(10), NPCON

Routines Using /SEGJ/

MAIN, CABC, CMNGF, CMSET, CMSW, CMWS, CMWW, CONECT, QDSRC, SFLDS, TBF, TRIO

/SEGJ/ is used to store the parameters defining current expansion functions. The equations for the current expansion functions are given in Section III-1 of Part I. The i^{th} current expansion function consists of a center section on segment i and branches on each segment connected to segment i. On segment j, where j is i or the number of a segment connected to segment i, the i^{th} expansion function is

$$f_j^i(s) = A_j^i + B_j^i \sin [k(s - s_j)] + C_j^i \cos [k(s - s_j)]$$

with the constants defined in Part I to match conditions on the current. A superscript i has been added to indicate the number of the current expansion function.

When subroutine TBF is called for expansion function i, it locates each segment connected to segment i and stores the segment number, j, in array JCO. TBF also computes the constants A_j^i, B_j^i, and C_j^i for segment j and stores them in AX, BX, and CX, respectively.

After all connected segments have been found, i is stored in the next location in JCO, and A_i^i, B_i^i, and C_i^i are stored in the corresponding locations in AX, BX, and CX.

/SEGJ/ is also used by subroutine TRIO. When TRIO is called for segment j, it locates each segment i connected to segment j and stores i in array JCO. TRIO calls SBF to compute the constants A_j^i, B_j^i, and C_j^i for the branch of expansion function i that extends onto segment j and stores these in AX, BX, and CX. The total number of entries, including i = j, is stored in JSNO. The remaining parameters are used with the NGF solution.

ISCON(I) = number of the segment in the NGF file having equation number I in the set of equations for modified basis functions. This is used when a new segment or patch connects to the NGF segment

NSCON = number of entries in ISCON

IPCON(I) = number of the patch in the NGF file having equation number I in the set of equations for modified patch basis functions. This is used when a new segment connects to the NGF patch

NPCON = number of entries in IPCON

<u>COMMON/SMAT/</u> SSX(16,16)

Routines Using /SMAT/

CMSET, FBLOCK GFIL, GFOUT, SOLVES

The array SSX is described under subroutine FBLOCK. In some copies of NEC-2 the variable name S is used in FBLOCK rather than SSX.

COMMON/TMH/ ZPK, RHKS

Routines Using /TMH/

GH, HFK

/TMH/ is used to pass values from HFK to GH. The variables ZPK and RHKS are defined in the discussion of subroutine HFK.

COMMON/TMI/ ZPK, RKB2, IJX

Routines Using /TMI/

EKSC, EKSCX, GF

/TMI/ is used to pass values from EKSC or EKSCX to GF. The meanings of the variables are listed in subroutines EKSC and EKSCX.

COMMON/VSORC/ VQD(10), VSANT(10), VQDS(10), IVQD(10), ISANT(10), IQDS(10), NVQD, NSANT, NQDS

Routines Using /VSORC/

MAIN, CABC, COUPLE, ETMNS, NETWK, QDSRC

The arrays in /VSORC/ contain the strengths and locations of voltage sources on wires. Separate arrays are used for applied-field voltage sources and current-derivative discontinuity voltage sources. The variables are defined as follows.

ISANT(I) = number of the segment on which the I^{th} applied-field source is located

IVQD(I) = IQDS(I) = number of the segment on end 1 of which the I^{th} current-slope discontinuity voltage source is located

VQD(I) = VQDS(I) = voltage of the I^{th} current-slope discontinuity source

VSANT(I) = voltage of the I^{th} applied-field voltage source

NSANT = number of applied-field voltage sources

NVQD = NQDS = number of current-slope discontinuity voltage sources

NVQD, IVQD, and VQD are set in MAIN from the input data. NQDS, IQDS, and VQDS are set in subroutine QDSRC. The latter were included to allow for current-slope discontinuities other than voltage sources, such as lumped loads. Loading by this means has not been implemented in NEC-2 however.

COMMON/YPARM/ NCOUP, ICOUP, NCTAG(5), NCSEG(5), Y11A(5), Y12A(20)

Routines Using /YPARM/

MAIN, COUPLE

Symbol Definitions:

NCOUP = number of segments between which coupling will be computed

ICOUP = number of segments in the coupling array that have been excited. When ICOUP = NCOUP subroutine COUPLE completes the coupling calculation

NCTAG(I) = tag number of segment I

NCSEG(I) = number of segment in set of segments having tag NCTAG(I)

Y11A(I) = self admittance of I^{th} segment specified

Y12A(I) = mutual admittances stored in order (1,2), (1,3), ... (2,3), (2,4), ... etc.

COMMON/ZLOAD/ ZARRAY(300)

Routines Using /ZLOAD/

MAIN, CMNGF, CMSET, GFIL, GFOUT, LOAD, QDSRC

ZARRAY(I) = $Z_I/(\Delta_I/\lambda)$, where Z_I is the total impedance on segment I, Δ_I is the length of segment I, and λ is the wavelength.

Section IV
System Library Functions Used by NEC

ABS(X)	= absolute value of X
AIMAG(Z)	= imaginary part of the complex number Z; result is real
AINT(X)	= integer truncation; result is real
ALOG(X)	= natural log of X
ALOG10(X)	= log to the base ten of X
ASIN(X)	= arcsine of X; result in radians
ATAN(X)	= arctangent of X; result in radians
ATAN2(X_1,X_2)	= arctangent of X_1/X_2; result in radians covering all four quadrants
CABS(Z)	= magnitude of the complex number, Z
CEXP(Z)	= complex exponential (e^Z)
CMPLX(X_1,X_2)	= formation of a complex number, $Z = X_1 + jX_2$
CONJG(Z)	= conjugate of the complex number Z
COS(X)	= cosine of X
CSQRT(Z)	= square root of a complex number, $\sqrt{Z}$
FLOAT(K)	= real number equivalent of integer K
IABS(K)	= absolute value of integer K
INT(X)	= X truncated to an integer
REAL(Z)	= real part of the complex number Z
SIN(X)	= sine of X
SQRT(X)	= square root of X
TAN(X)	= tangent of X

Section V
Array Dimension Limitations

Array dimensions in the program limit the structure model in various ways. Any of these limits may be increased if necessary at the expense of core storage capacity, which may require reducing other array dimensions. The limits imposed by array dimensions are described below.

In-Core Matrix Storage, I_r = 4000.

Arrays:

COMMON/CMB/ CM(I_r)

Limit constant:

IRESRV = I_r at MA68 of MAIN

I_r is the number of words of core available for storage of the interaction matrix. The complete matrix will fit in core storage if (N + 2M) × (NP + 2MP) is not greater than I_r. For out-of-core solution, I_r must be at least 2(N + 2M) and should be as large as possible to minimize file manipulation.

Maximum Segments and Patches

Minimum Dimensions for N segments and M patches:

COMMON/DATA/ X(N + M), Y(N + M), Z(N + M), SI(N + M), BI(N + M), ALP(N + M), BET(N + M), ICON1(N + M), ICON2(N + M), ITAG(N + M), ICONX(N + M)

COMMON/CRNT/AIR (N), AII(N), BIR(N), BII(N), CIR(N), CII(N), CUR(N + 3M)

COMMON/ANGL/ SALP(N + M)

COMMON/SAVE/ IP(N + 2M)

COMMON/ZLOAD/ ZARRAY(N)

COMMON/SCRATM/ D(N + 2M) or Y(N + 2M)

MAIN: IX(N + 2M)

SUBROUTINE NETWK: RHS(N + 3M)

Limit Constant:

LD = N + M at MA66 of MAIN

All segments and patches resulting from reflection or rotation of a symmetric structure must be included in determining the limiting structure size.

Maximum Number of Non-radiating Networks, N_n = 30.

Arrays:

COMMON/NETCX/: ISEG1(N_n), ISEG2(N_n), X11R(N_n), X11I(N_n), X12R(N_n), X12I(N_n), X22R(N_n), X22I(N_n), NTYP(N_n)

SUBROUTINE NETWK: RHNT(N_n), IPNT(N_n), NTEQA(N_n), NTSCA(N_n), RHNX(N_n), CMN(N_n,N_n)

Limit Constants:

NETMX = N_n at MA63 of MAIN

NDIMN = N_n at NT22 of NETWK

NDIMNP = N_n + 1 at NT22 of NETWK

N_n is the limit for either the number of networks (including transmission lines) or the number of segments having one or more network ports connected, whichever is greater. When relative driving point matrix asymmetry is computed, N_n must also be greater than or equal to the sum of the number of segments with network ports connected plus the number of segments with voltage sources.

Maximum Number of Degrees of Symmetry, $N_p = 16$.

Arrays:

COMMON/SMAT/ $S(N_p, N_p)$

N_p limits the number of symmetric cells in a structure. The number of symmetric cells is equal to the ratio of N to NP in COMMON/DATA/.

Maximum Number of Segments Joined at Junctions, $N_j = 30$

If N^- and N^+ are the numbers of segments connected to end 1 and end 2 of a segment, respectively, then the dimensions in COMMON/SEGJ/, N_j, must be at least $N^- + N^+ + 1$.

Array:

COMMON/SEGJ/ $AX(N_j)$, $BX(N_j)$, $CX(N_j)$, $JCO(N_j)$, JSNO

Limit Constants:

JMAX = N_j at SB6 in SBF
JMAX = N_j at TB8 in TBF
JMAX = N_j at TR8 in TRIO

Maximum Number of Voltage Sources, $N_v = 30$.

Arrays:

COMMON/VSORC/ $VQD(N_v)$, $VSANT(N_v)$, $VQDS(N_v)$, $IVQD(N_v)$, $ISANT(N_v)$, $IQDS(N_v)$

Limit Constant:

NSMAX = N_v at MA63 of MAIN

A model may use up to N_v applied field voltage sources and up to N_v current slope discontinuity voltage sources.

Maximum Number of Loading Cards, $N_1 = 30$

Arrays:

MAIN: LDTYP(N_1), LDTAG(N_1), LDTAGF(N_1), LDTAGT(N_1), ZLR(N_1), ZLI(N_1), ZLC(N_1)

Limit Constants:

LOADMX = N_1 at MA63 of MAIN

When the NGF option is used only new loading cards are counted, not those used in generating the NGF file.

Number of Comment Cards Saved, $N_c = 5$

Arrays:

COMMON/SAVE/: COM(13,N_c)

Limit Constant:

Constants at MA71 of MAIN

Any number of comment cards may be placed at the beginning of a data deck and will be printed in the output. Only N_c of the cards will be saved in array COM for later use in labeling plots, however. The first $N_c - 1$ comment cards and the last comment card will be saved.

Maximum Field Points for Normalized Gain, $N_g = 1200$.

Arrays:

COMMON/SCRATM/ GAIN(N_g)

Limit Constant:

NORMAX = N_g at RD22 of SUBROUTINE RDPAT

N_g is the maximum number of field points from a single RP data card that can be stored for output in normalized form or for plotting if plotting is

implemented. If an RP card requesting more than N_g points calls for normalized gain, the gain will be computed and printed at all requested angles, but only the first N_g gains will be stored and normalized.

COMMON/SCRATM/ GAIN occurs in SUBROUTINE RDPAT. COMMON/SCRATM/ D and COMMON/SCRATM/ Y occur in certain other routines where D and Y are complex (see "Maximum Segments and Patches"). GAIN, D, and Y should be dimensioned so that each common statement contains the same number of words.

<u>Maximum Number of Frequencies for Normalized Impedance or Maximum Number of Angles for Which Received Signal Strength Is Stored, N_f = 200</u>

Array:

MAIN: FNORM(N_f)

Limit Constant:

NORMF = N_f at MA63 of MAIN

The maximum number of frequencies for which input impedance may be stored and normalized is $N_f/4$, since the real and imaginary impedance and magnitude and phase are each stored. The receiving current can be stored for up to N_f angles.

Maximum Number of Points in Coupling Calculation, $N_c = 5$.

The maximum number of segments among which coupling can be computed (CP cards) is N_c.

COMMON/YPARM/: NCTAG(N_c), NCSEG(N_c), Y11A(N_c), Y12A($N_c^2 - N_c$)

Limit Constants:

Constants at MA207 and MA212 of MAIN should equal N_c

Maximum Number of NGF Segments to Which New Segments or Patches Connect, $N_s = 50$

COMMON/SEGJ/: ISCON(N_s)

Limit Constant:

NSMAX = N_s at CN13 of CONECT

Maximum Number of NGF Patches to Which New Segments Connect, $N_p = 10$.

COMMON/SEGJ/: IPCON(N_p)

Limit Constant:

NPMAX = N_p at CN13 of CONECT

Section VI
Overview of Numerical Green's Function Operation

NEC includes a provision to generate and factor an interaction matrix and save the result on a file. A later run, using the file, may add to the structure and solve the complete model without unnecessary repetition of calculations. This procedure is called the Numerical Green's Function (NGF) option since the effect is as if the free space Green's function in NEC were replaced by the Green's function for the structure on the file. The NGF is particularly useful for a large structure, such as a ship, on which various antennas will be added or modified. It also permits taking advantage of partial symmetry since a NGF file may be written for the symmetric part of a structure, taking advantage of the symmetry to reduce computation time. Unsymmetric parts can then be added in a later run.

For the NGF solution the matrix is partitioned as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} ,$$

where A is the interaction matrix for the initial structure, D is the matrix for the added structure, and B and C represent mutual interactions. The current is computed as

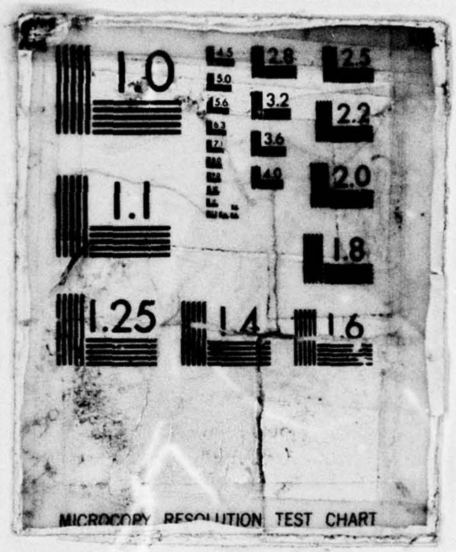
$$I_2 = \left[D - CA^{-1}B\right]^{-1} \left[E_2 - CA^{-1}E_1\right] ,$$

$$I_1 = A^{-1}E_1 - A^{-1}BI_2 ,$$

after the factored matrix A has been read from the NGF file along with other necessary data. Since the LU decomposition is obtained in NEC rather than the inverse, the multiplication by A^{-1} is accomplished by using the solution procedure in subroutine SOLVE on each column in the matrix to the right of A^{-1}.

To use the NGF option the parameters of the fixed, or NGF, part of the model are defined in the first run. A WG data card causes the matrix A to be computed (CMSET), factored (FACTRS), and written to file TAPE20 by subroutine

AD-A075 289 NUMERICAL ELECTROMAGNETIC CODE (NEC)-METHOD OF MOMENTS 6/6
A USER-ORIENTED CO.. (U) NAVAL OCEAN SYSTEMS CENTER SAN
DIEGO CA G J BURKE ET AL. 18 JUL 77 NOSC/TD-116-VOL-1
UNCLASSIFIED AFWL-TR-76-320-VOL-1 F/G 12/5 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

GFOUT. Other necessary data, such as segment and patch coordinates, frequency, loading, and ground parameters, are also written to TAPE20.

When the NGF file, TAPE20, is used the data are read into the usual arrays by subroutine GFIL and new segments and patches are added to the arrays in COMMON/DATA/. Subroutine CMNGF is then called to compute the matrix elements in B, C, and D. FACGF computes $A^{-1}B$, storing it in place of B, and computes $(D - CA^{-1}B)$, factors it into L and U parts, and stores the result in place of D. For each excitation E_1 and E_2, SOLGF completes the procedure of solving for I_1 and I_2.

The procedure is complicated by the connection of new segments or patches to NGF segments or patches. A connection to a segment modifies the current basis function (see Section III.1 of Part I). Since the elements in A cannot be changed, a modified basis function must be treated as a new basis function with a new column added to B and D and the new basis function amplitude added to the end of I_2. The amplitude of the original basis function is set to zero by adding a row containing all zeros except for a one in the column of C corresponding to the modified basis function. Since the segment is not modified the boundary condition equation is not altered in this case.

When a new segment connects to a NGF patch the patch must be divided into four new patches, after the user defined patches, requiring eight new rows and columns in B, C, and D. Two additional rows are added to set the two current components on the old patch to zero. Since the old patch is replaced by the four new patches, the condition on the field at the center of the patch should be removed. This is done by adding two new columns each containing all zeros except for a one in the row of the equation to be removed.

The matrix structure is further complicated by the division of each submatrix into sections for segment-to-segment, patch-to-patch, segment-to-patch and patch-to-segment interactions. The matrix structure is shown in Figure 12, where the subscript w denotes wire segments and s denotes surface patches. The elements of B'_{ww} and B'_{sw} are the E fields and H fields due to modified basis functions in the NGF section. Each column of B'_{ss} and row of C'_{ww} and C'_{ss} contains 0's and a single 1.

The subroutine ETMNS fills the excitation array with the E fields illuminating all segments, followed by the H fields on patches. These elements are reordered in SOLGF to correspond to the matrix structure. After

A_{ww}	A_{ws}	B_{ww}	B_{ws}	B'_{ww}	0
A_{sw}	A_{ss}	B_{sw}	B_{ss}	B'_{sw}	B'_{ss}
C_{ww}	C_{ws}	D_{ww}	D_{ws}	D'_{ww}	0
C_{sw}	C_{ss}	D_{sw}	D_{ss}	D'_{sw}	0
C'_{ww}	0	0	0	0	0
0	C'_{ss}	0	0	0	0

Figure 12. Matrix Structure for the NGF Solution

the solution this reordering is reversed in SOLGF to put basis function amplitudes for segments first, followed by those for patches. If symmetry is used in the NGF section the matrix A is structured as submatrices for the symmetric sections. Each submatrix contains elements for segments and patches in that section, with the order as shown for A in Figure 12. In this case the excitation and solution vectors are ordered in SOLVES to correspond to the submatrix structure.

Section VII
Overview of Matrix Operations Using File Storage

File storage is used when the matrix size exceeds the length of the array CM in COMMON/CMB/. For the basic solution (not NGF) there are five matrix storage modes associated with the integer ICASE as follows:

ICASE	Matrix Storage
1	Matrix fits in CM; no structure symmetry
2	Matrix fits in CM; structure symmetry used
3	Matrix stored on file; no symmetry
4	Matrix stored on file; symmetry; each submatrix fits into CM for LU decomposition
5	Matrix stored on file; symmetry; submatrices do not fit into CM.

For case 3 the matrix is initially written on file 11 by blocks of rows. The block size is chosen in subroutine FBLOCK so that two blocks will fit into CM for the Gauss elimination procedure. The block size and number of blocks is set by the parameters NBLOKS, NPBLK, and NLAST in COMMON/MATPAR/.

Subroutine FACIO reads file 11 and writes file 12 using 13 and 14 for scratch storage. LUNSCR then reads 12 and writes the blocks of the factored matrix on file 13 in forward order and on file 14 in reversed order. File 13 is then used for forward substitution in the solution and file 14 is used for backward substitution.

For case 4, FACTRS reads the matrix from file 11, where it was written by blocks of rows (columns of the transposed matrix), and writes it to file 12 by submatrices. The submatrices are then read from 12, factored, and written to 13.

In case 5, FACTRS reads the matrix from file 11 and writes it to file 12 by blocks of rows (columns of the transposed matrix) for each submatrix. File 12 is then copied back to file 11, and the procedure of case 3 is repeated for each submatrix.

When a NGF file is to be written, half of CM is reserved for matrix storage and manipulations of the matrices B, C, and D. Hence for cases 1, 2 or 4 the primary matrix A (or submatrix for case 4) must fit into half of CM.

There is no restriction for cases 3 or 5 since, with two matrix blocks fitting into CM for the LU decomposition, half of CM is available during the solution when blocks are used one at a time.

There are four modes for storing B, C, and D in the NGF solution. These are associated with the integer ICASX as follows:

A_F = matrix A factored into L and U

$$A_R = \begin{cases} A_F \text{ for ICASE} = 1 \text{ or } 2 \\ \text{one block of } A_F \text{ for ICASE} = 3 \\ \text{one submatrix for ICASE} = 4 \\ \text{one block of submatrix for ICASE} = 5 \end{cases}$$

A_X = A_F for ICASE = 1 or 2
nothing otherwise

ICASX	NGF Matrix Storage
1	A_R, B, C, and D fit into CM
2	B, C, and D fit into CM but not with A_R (ICASE = 3, 4, 5) A_R and B must also fit into CM together
3	B, C, and D do not fit into CM, but A_X and $F = D - CA^{-1}B$ fit into CM for the LU decomposition of F
4	Same as 3 but $D - CA^{-1}B$ requires file storage for LU decomposition

When a NGF file (TAPE20) is written with ICASE = 3 or 5, files 13 and 14 are both written to TAPE20. When the NGF file is read these data are written on the single file 13 with the blocks in ascending order first and then in descending order. If A_F is stored on file 13 then space for A_R in CM is needed only when A_R is used in a solution in CM. This accounts for the definition of A_X.

File usage for ICASX = 2, 3, and 4 is outlined in Figures 13 and 14. The value for ICASX is chosen in subroutine FBNGF as the smallest value possible. The number of blocks into which matrices B, C, and D are divided is also chosen in FBNGF.

NGF Procedure for ICASX = 2	Contents of CM	Files 11	12	13	14	15	16
				A_F			
1. (CMNGF) Compute B, C and D in CM. Write to files 12, 14 and 15.	B, C, D		D	A_F	B	C	
2. (FACGF) Read 13 and 14. Compute $A^{-1}B$. Write 14.	A_F, B		D	A_F	$A^{-1}B$	C	
3. Compute $F = D - CA^{-1}B$. Store over D in CM.	$A^{-1}B$, C, D		D	A_F	$A^{-1}B$	C	
4. Factor F. Write on 11.	$A^{-1}B$, C, F	F_F		A_F	$A^{-1}B$	C	
Solution for excitation $(E_1, E_2)^T$ (SOLGF)							
1. Compute $I_1' = A^{-1}E_1$	A_F	F_F		A_F	$A^{-1}B$	C	
2. $I_2' = E_2 - CA^{-1}E_1 = E_2 - CI_1'$	C						
3. $I_2 = F^{-1}I_2'$	F_F						
4. $I_1 = I_1' - (A^{-1}B)I_2$	$A^{-1}B$						

(Subscript F indicates that the matrix has been factored into L and U triangular parts)

Figure 13. NGF File Usage for ICASX = 2.

NGF Procedure for ICASX = 3,4	Contents of CM	11	12	13	14	15	16
				A_F			
1. (CMNGF) Compute B by blocks of rows. Write to file 14.	A_X, B_R			A_F	B_R		
2. Compute C and D by blocks by rows. Write to 15 and 12.	A_X, C_R, D_R		D_R	A_F	B_R	C_R	
3. (REBLK) Read 14. Write B by blocks of columns on file 16.	A_X, B_C, B_R		D_R	A_F	B_R	C_R	B_C
4. (FACGF) Read 16; compute $A^{-1}B$; write 14.	A_F, B_C		D_R	A_F	$A^{-1}B_C$	C_R	
5. Read blocks from 12, 14 and 15 and compute $F = D - C(A^{-1}B)$ by blocks of rows. Write on 11.	A_X, $A_{-1}B_C$ C_R, D_R	F	D_R	A_F	$A^{-1}B_C$	C_R	
6. For ICASX = 4 call FACIO to factor F. FACIO reads 11 and writes 12, using 11 and 16 as scratch storage.	A_X and 2 blocks of F	X	F_F'	A_F	$A^{-1}B_C$	C_R	X
7. LUNSCR reads 12 and writes blocks of FF on 111 in forward order and on 16 in reversed order.	A_X and block of F	F_F		A_F	$A^{-1}B_C$	C_R	F_F
6' For ICASX = 3, read all blocks of F into CM; Factor F; write to 11.	A_X, F	F_F		A_F	$A^{-1}B_C$	C_R	

Figure 14. NGF File Usage for ICASX = 3 or 4.

Section VIII
NEC Subroutine Linkage

Figures 15 and 16 show the organization of subroutines in the NEC-2 program. All possible subroutine calls are traced, although in a particular run only certain of the traces will be followed. Routines that are called at more than one point in the program are shown as separate blocks for each call.

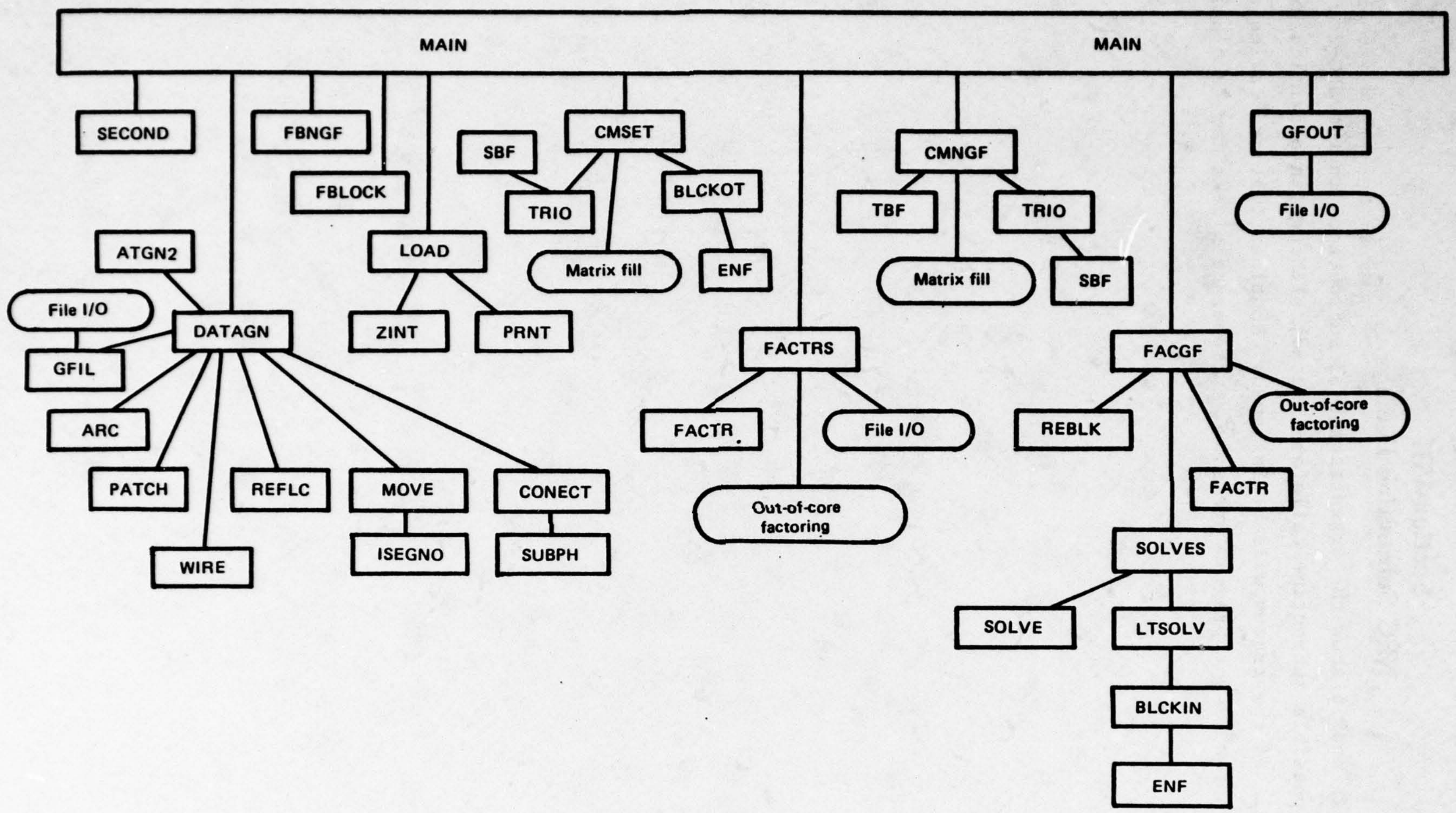


Figure 15. NEC Subroutine Linkage Chart. For Block Definitions, see Figure 16

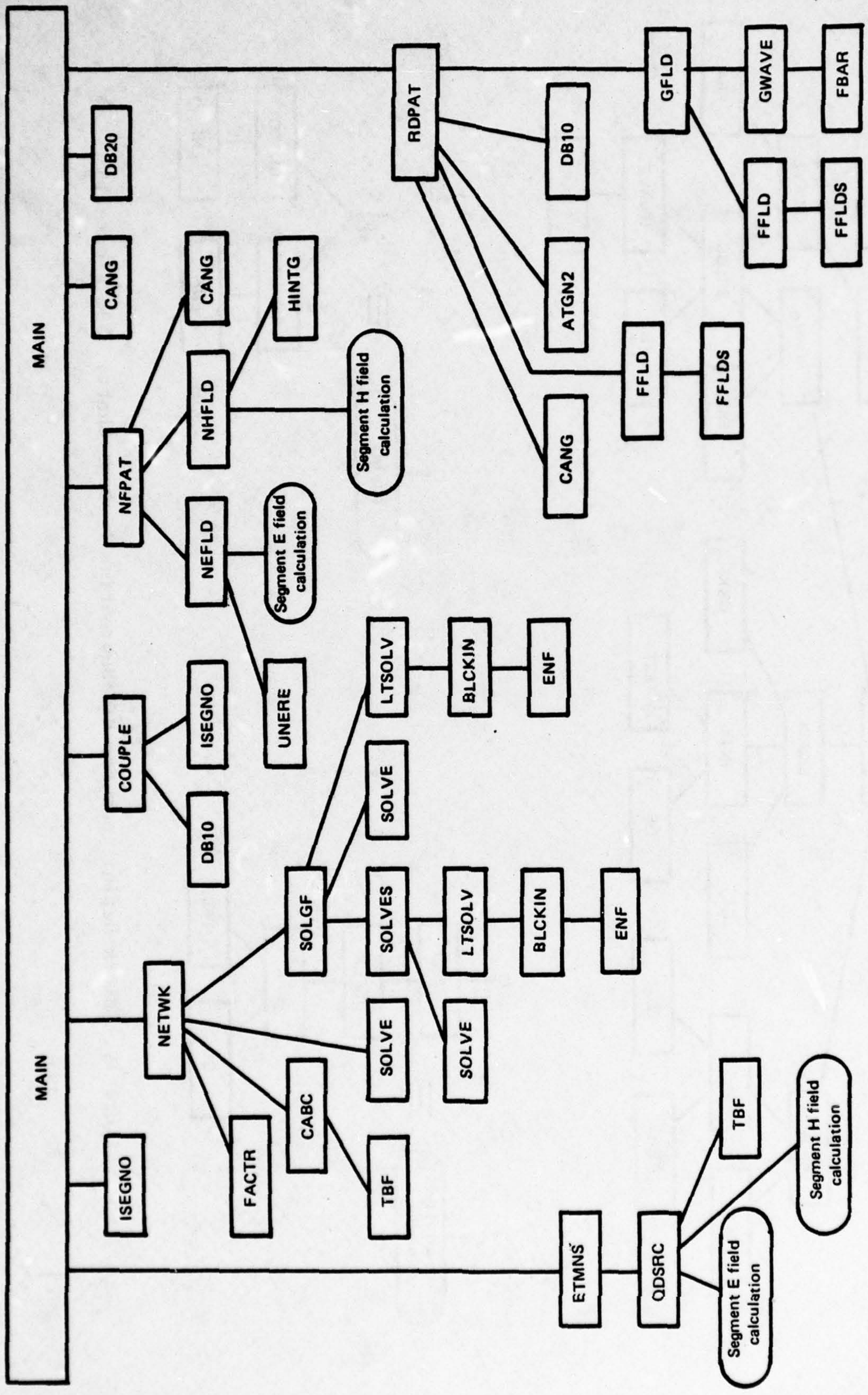


Figure 15 (continued)

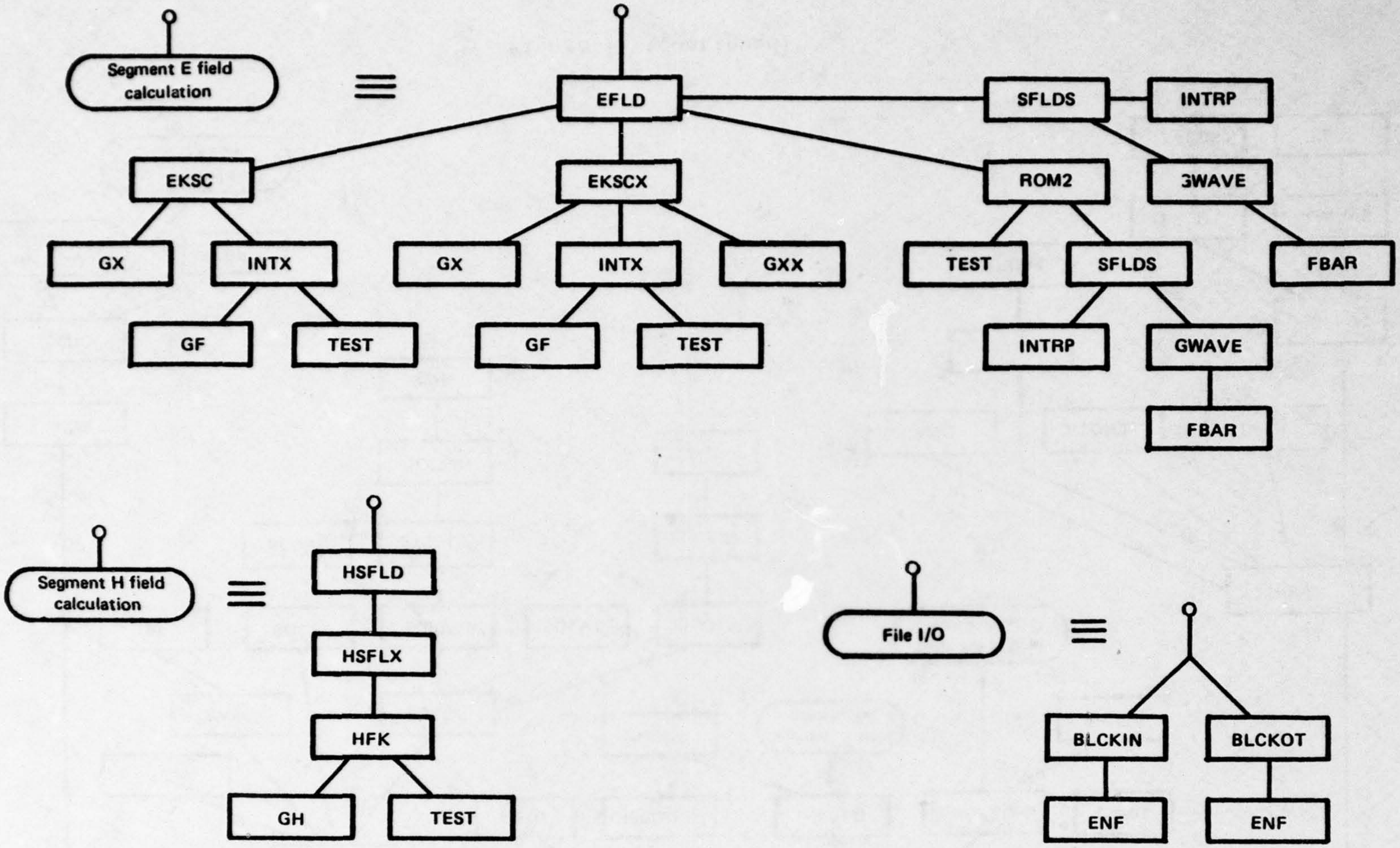


Figure 16. Block Definitions for NEC Subroutine Linkage Chart. See Figure 15

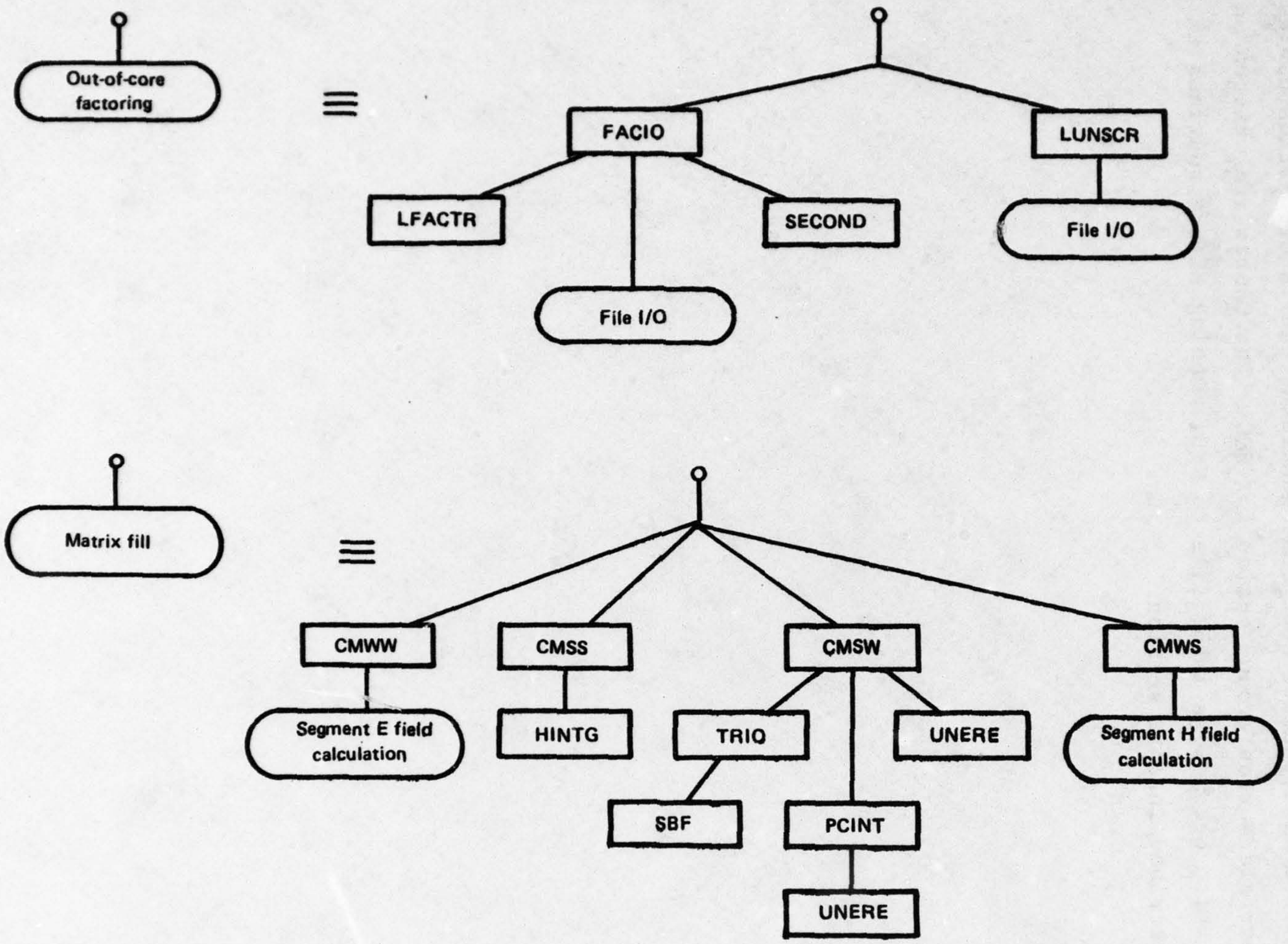


Figure 16 (continued)

Section IX
SOMNEC

1. SOMNEC CODE DESCRIPTION

SOMNEC is an independent code that generates the interpolation tables for the Sommerfeld/Norton ground option for NEC. The tables are written on file TAPE21 which becomes an input file to NEC. Coding of the routines in SOMNEC is described in this section.

SOMNEC (main program)

PURPOSE

To generate interpolation tables for the Sommerfeld/Norton ground option and write them on file TAPE21.

METHOD

The code from SN17 to SN51 reads the input data and sets parameters in COMMON/EVLCOM/. Since all equations are scaled to a free-space wavelength of one meter the results depend only on the complex dielectric constant

$$\epsilon_c = \epsilon_1 - j\sigma_1/(\omega\epsilon_0) .$$

In the routines that evaluate the Sommerfeld integrals the time dependence is exp(-jωt) rather than exp(+jωt) which is used in the remainder of NEC. Hence the conjugate of ϵ_c (EPSCF) is taken before computing the parameters in COMMON/EVLCOM/. The conjugate of the results is taken at the end of EVLUA, so the results returned to SOMNEC and written on TAPE21 are for exp(+jωt).

Three interpolation tables, as shown in Figure 12 of Part I, are generated in the code from SN55 to SN123. For each R_1, θ pair in the tables the values of ρ and z + z' are computed and stored in COMMON/EVLCOM/. Subroutine EVLUA is then called and returns the quantities

$$\mathrm{ERV} = \frac{\partial^2}{\partial\rho\partial z} k_1^2 V_{22}'$$

$$\mathrm{EZV} = \left(\frac{\partial^2}{\partial z^2} + k_2^2\right) k_1^2 V_{22}'$$

$$\mathrm{ERH} = \left(\frac{\partial^2}{\partial\rho^2} k_2^2 V_{22}' + k_2^2 U_{22}'\right)$$

$$\mathrm{EPH} = -\left(\frac{1}{\rho}\frac{\partial}{\partial\rho} k_2^2 V_{22}' + k_2^2 U_{22}'\right)$$

These are multiplied by C_1 R_1 exp(jkR_1) to form the quantities in equation (156) through (159) in Part I. When R_1 is zero the limiting forms in equations (169) through (172) of Part I are used. The expressions from

SN116 to SN118 are obtained by letting θ go to zero in the expressions for $R_1 = 0$.

The data are stored in COMMON/GGRID/ which is identical to the common block in NEC. File 21 is written at SN127 and includes coordinates of the grid boundaries, number of points, and increments for R_1 and θ. Hence those grid parameters can be changed in SOMNEC without changing NEC. If the number of grid points is increased, however, the arrays in COMMON/GGRID/ must be increased in both SOMNEC and NEC. Also, the parameters NDA and NDPA in subroutine INTRP must be changed.

SYMBOL DICTIONARY

AR1 = array for grid 1

AR2 = array for grid 2

AR3 = array for grid 3

CK1 = k_1

CK1R = real part of k_1

CK1SQ = k_1^2

CK2 = k_2 (= 2π since $\lambda = 1$)

CK2SQ = k_2^2

CKSM = $k_2^2/(k_1^2 + k_2^2)$

CL1 = $k_2^2 C_1 C_3$ (see Part 1 for C_1, C_2, and C_3)

CL2 = $k_2^2 C_1 C_2$

CON = $C_1 R_1 \exp(jkR_1)$

CT1 = $(k_1^2 - k_2^2)/2$

CT2 = $(k_1^4 - k_2^4)/8$

CT3 = $(k_1^6 - k_2^6)/16$

DR = ΔR_1

DTH = $\Delta\theta$

DXA = ΔR_1 for each grid

DYA = $\Delta\theta$ for each grid (radians)

EPH = EPH

EPR = ϵ_1

EPSCF = ϵ_c

ERH = ERH

ERV = ERV

EZV = EZV

FMHZ = frequency in MHz

IPT = flag to control printing of grid

IR = index for R_1 values

IRS = starting value for IR

ITH = index for θ values

LCOMP = labels for output

NR = number of R_1 values

NTH = number of θ values

NXA = number of R_1 values for each grid

NYA = number of θ values for each grid

R = R_1

RHO = ρ

RK = k_2R

SIG = σ_1

TFAC1 = $(1 - \sin\theta)/\cos\theta$

TFAC2 = $(1 - \sin\theta)/\cos^2\theta$

THET = θ

TIM = time to fill arrays

TKMAG = $100.|k_1|$

TSMAG = $100.|k_1|^2$

TST = starting time

WLAM = wavelength in free space

XSA = starting value of R_1 in each grid

YSA = starting value of θ in each grid

ZPH = $Z + Z'$

CONSTANTS

299.8 = 10^{-6} times velocity of light in m/s

59.96 = $1/(2\pi c\epsilon_o)$, c = velocity of light

6.283185308 = 2π

```
      PROGRAM SOMNEC(INPUT,OUTPUT,TAPE21)
C
C     PROGRAM TO GENERATE NEC INTERPOLATION GRIDS FOR FIELDS DUE TO
C     GROUND.  FIELD COMPONENTS ARE COMPUTED BY NUMERICAL EVALUATION
C     OF MODIFIED SOMMERFELD INTEGRALS.
C
      COMPLEX CK1,CK1SQ,ERV,EZV,ERH,EPH,AR1,AR2,AR3,EPSCF,CKSM,CT1,CT2,C
     1T3,CL1,CL2,CON
      COMMON /EVLCOM/ CKSM,CT1,CT2,CT3,CK1,CK1SQ,CK2,CK2SQ,TKMAG,TSMAG,C
     1K1R,ZPH,RHO,JH
      COMMON /GGRID/ AR1(11,10,4),AR2(17,5,4),AR3(9,8,4),EPSCF,DXA(3),DY
     1A(3),XSA(3),YSA(3),NXA(3),NYA(3)
      DIMENSION LCOMP(4)
      DATA NXA/11,17,9/,NYA/10,5,8/,XSA/0.,.2,.2/,YSA/0.,0.,.3490658504/
      DATA DXA/.02,.05,.1/,DYA/.1745329252,.0872664626,.1745329252/
      DATA LCOMP/3HERV,3HEZV,3HERH,3HEPH/
C
C     READ GROUND PARAMETERS - EPR = RELATIVE DIELECTRIC CONSTANT
C                              SIG = CONDUCTIVITY (MHOS/M)
C                              FMHZ = FREQUENCY (MHZ)
C                              IPT = 1 TO PRINT GRIDS.  =0 OTHERWISE.
C     IF SIG .LT. 0. THEN COMPLEX DIELECTRIC CONSTANT = EPR + J*SIG
C     AND FMHZ IS NOT USED
C
      READ 15, EPR,SIG,FMHZ,IPT
      IF (SIG.LT.0.) GO TO 1
      WLAM=299.8/FMHZ
      EPSCF=CMPLX(EPR,-SIG*WLAM*59.96)
      GO TO 2
1     EPSCF=CMPLX(EPR,SIG)
2     CALL SECOND (TST)
      CK2=6.283185308
      CK2SQ=CK2*CK2
C
C     SOMMERFELD INTEGRAL EVALUATION USES EXP(-JWT), NEC USES EXP(+JWT),
C     HENCE NEED CONJG(EPSCF).  CONJUGATE OF FIELDS OCCURS IN SUBROUTINE
C     EVLUA.
C
      CK1SQ=CK2SQ*CONJG(EPSCF)
      CK1=CSQRT(CK1SQ)
      CK1R=REAL(CK1)
      TKMAG=100.*CABS(CK1)
      TSMAG=100.*CK1*CONJG(CK1)
      CKSM=CK2SQ/(CK1SQ+CK2SQ)
      CT1=.5*(CK1SQ-CK2SQ)
      ERV=CK1SQ*CK1SQ
      EZV=CK2SQ*CK2SQ
      CT2=.125*(ERV-EZV)
      ERV=ERV*CK1SQ
      EZV=EZV*CK2SQ
      CT3=.0625*(ERV-EZV)
C
C     LOOP OVER 3 GRID REGIONS
C
      DO 6 K=1,3
      NR=NXA(K)
      NTH=NYA(K)
      DR=DXA(K)
      DTH=DYA(K)
      R=XSA(K)-DR
      IRS=1
      IF (K.EQ.1) R=XSA(K)
      IF (K.EQ.1) IRS=2
```

```
C
C     LOOP OVER R.  (R=SQRT(RHO**2 + (Z+H)**2))
C
      DO 6 IR=IRS,NR
      R=R+DR
      THET=YSA(K)-DTH
C
C     LOOP OVER THETA.  (THETA=ATAN((Z+H)/RHO))
C
      DO 6 ITH=1,NTH
      THET=THET+DTH
      RHO=R*COS(THET)
      ZPH=R*SIN(THET)
      IF (RHO.LT.1.E-7) RHO=1.E-8
      IF (ZPH.LT.1.E-7) ZPH=0.
      CALL EVLUA (ERV,EZV,ERH,EPH)
      RK=CK2*R
      CON=-(0.,4.77147)*R/CMPLX(COS(RK),-SIN(RK))
      GO TO (3,4,5), K
3     AR1(IR,ITH,1)=ERV*CON
      AR1(IR,ITH,2)=EZV*CON
      AR1(IR,ITH,3)=ERH*CON
      AR1(IR,ITH,4)=EPH*CON
      GO TO 6
4     AR2(IR,ITH,1)=ERV*CON
      AR2(IR,ITH,2)=EZV*CON
      AR2(IR,ITH,3)=ERH*CON
      AR2(IR,ITH,4)=EPH*CON
      GO TO 6
5     AR3(IR,ITH,1)=ERV*CON
      AR3(IR,ITH,2)=EZV*CON
      AR3(IR,ITH,3)=ERH*CON
      AR3(IR,ITH,4)=EPH*CON
6     CONTINUE
C
C     FILL GRID 1 FOR R EQUAL TO ZERO.
C
      CL2=-(0.,188.370)*(EPSCF-1.)/(EPSCF+1.)
      CL1=CL2/(EPSCF+1.)
      EZV=EPSCF*CL1
      THET=-DTH
      NTH=NYA(1)
      DO 9 ITH=1,NTH
      THET=THET+DTH
      IF (ITH.EQ.NTH) GO TO 7
      TFAC2=COS(THET)
      TFAC1=(1.-SIN(THET))/TFAC2
      TFAC2=TFAC1/TFAC2
      ERV=EPSCF*CL1*TFAC1
      ERH=CL1*(TFAC2-1.)+CL2
      EPH=CL1*TFAC2-CL2
      GO TO 8
7     ERV=0.
      ERH=CL2-.5*CL1
      EPH=-ERH
8     AR1(1,ITH,1)=ERV
      AR1(1,ITH,2)=EZV
      AR1(1,ITH,3)=ERH
9     AR1(1,ITH,4)=EPH
      CALL SECOND (TIM)
C
C     WRITE GRID ON TAPE21
C
      WRITE (21) AR1,AR2,AR3,EPSCF,DXA,DYA,XSA,YSA,NXA,NYA
```

```
      REWIND 21                                                         SN 128
      IF (IPT.EQ.0) GO TO 14                                            SN 129
C                                                                       SN 130
C     PRINT GRID                                                        SN 131
C                                                                       SN 132
      PRINT 17, EPSCF                                                   SN 133
      DO 13 K=1,3                                                       SN 134
      NR=NXA(K)                                                         SN 135
      NTH=NYA(K)                                                        SN 136
      PRINT 18, K,XSA(K),DXA(K),NR,YSA(K),DYA(K),NTH                    SN 137
      DO 13 L=1,4                                                       SN 138
      PRINT 19, LCOMP(L)                                                SN 139
      DO 13 IR=1,NR                                                     SN 140
      GO TO (10,11,12), K                                               SN 141
10    PRINT 20, IR,(AR1(IR,ITH,L),ITH=1,NTH)                            SN 142
      GO TO 13                                                          SN 143
11    PRINT 20, IR,(AR2(IR,ITH,L),ITH=1,NTH)                            SN 144
      GO TO 13                                                          SN 145
12    PRINT 20, IR,(AR3(IR,ITH,L),ITH=1,NTH)                            SN 146
13    CONTINUE                                                          SN 147
14    TIM=TIM-TST                                                       SN 148
      PRINT 16,TIM                                                      SN 149
      STOP                                                              SN 150
C                                                                       SN 151
15    FORMAT (3E10.3,I5)                                                SN 152
16    FORMAT (6H TIME=,E12.5)                                           SN 153
17    FORMAT (30H1NEC GROUND INTERPOLATION GRID,/,21H DIELECTRIC CONSTAN SN 154
     1T=,2E12.5)                                                        SN 155
18    FORMAT (///,5H GRID,I2,/,4X,5HR(1)=,F7.4,4X,3HDR=,F7.4,4X,3HNR=,I3 SN 156
     1,/,9H THET(1)=,F7.4,3X,4HDTH=,F7.4,3X,4HNTH=,I3,//)               SN 157
19    FORMAT (///,A3)                                                   SN 158
20    FORMAT (4H IR=,I3,/,(10E12.5))                                    SN 159
      END                                                               SN 160-
```

BESSEL

PURPOSE

To compute the Bessel function of order zero and its derivative for a complex argument.

METHOD

For argument magnitudes less than a limit Z_s the functions are evaluated by the ascending series and for larger magnitudes by Hankel's asymptotic expansion (ref. 5). The ascending series are

$$J_0(Z) = \sum_{k=0}^{\infty} \frac{(-Z^2/4)^k}{(k!)^2}$$

$$J_0'(Z) = -J_1(Z) = -\frac{Z}{2} \sum_{k=0}^{\infty} \frac{(-Z^2/4)^k}{k!(k+1)!}$$

The number of terms used with an argument Z is M(IZ) where IZ = 1. + $|Z|^2$. The array M is filled for IZ from 1 to 101 on the first call to BESSEL by determining the value of k at which the term in the series for J_0 is less than 10^{-6}.

When $|Z|$ is greater than Z_s Hankel's asymptotic expansions are used with two or three terms. These are

$$J_\nu(Z) = \sqrt{\frac{2}{\pi Z}}\,[P(\nu,Z)\cos\chi - Q(\nu,Z)\sin\chi] \qquad (|\arg Z| < \pi)$$

$$\chi = Z - (\tfrac{1}{2}\nu + \tfrac{1}{4})\pi$$

$$P(\nu,Z) = 1 - \frac{(\mu-1)(\mu-9)}{2!(8Z)^2} + \frac{(\mu-1)(\mu-9)(\mu-25)(\mu-49)}{4!(8Z)^4}$$

$$Q(\nu,Z) = \frac{(\mu-1)}{8} - \frac{(\mu-1)(\mu-9)(\mu-25)}{3!(8Z)^3}$$

where $\mu = 4\nu^2$.

When $Z_s < |Z| < Z_s + \Delta$ both the series and asymptotic forms are evaluated and are combined as

$$J(Z) = \frac{1}{2}\left[J_s(Z)(1+C) + J_a(Z)(1-C)\right]$$

where $C = \cos\left(\frac{\pi}{\Delta}(|Z| - Z_s)\right)$

$J_s(Z)$ = result of series evaluation

$J_a(Z)$ = result of asymptotic evaluation

This combination ensures a smooth transition between the two regions. In the code Z_s is 6 and Δ is 0.1.

SYMBOL DICTIONARY

A1	$= -1./(4k^2)$
A2	$= 1./(k+1)$
C3	$= \sqrt{2/\pi} = 0.7978845608$
CZ	$= \cos\chi$
FJ	$= \sqrt{-1}.$
FJX	= FJ
IB	= 1 to indicate that both the series and asymptotic forms will be evaluated and combined
INIT	= flag to indicate that initialization of constants has been completed
IZ	$= 1. + \|Z\|^2$ truncated to an integer
JO	$= J_0(Z)$
JOP	$= J_0'(Z)$
JOPX	$= J_0'(Z)$ from series to be combined with asymptotic result
JOX	$= J_0(Z)$, same as JOPX
K	= summation index k, summed from 1 to limit

M	=	array of upper limits for k
MIZ	=	upper limit for k
POZ	=	P(0,z)
P10	=	coefficient in POZ = $9/(2 \times 8^2)$
P11	=	coefficient in P1Z = $-(4 - 1)(4 - 9)/(2 \times 8^2)$
P1Z	=	P(1,Z)
P20	=	coefficient in POZ = $9 \times 25 \times 49/(4!8^4)$
P21	=	coefficient in P1Z = $-(4 - 1)(4 - 9)(4 - 25)(4 - 49)/(4!8^4)$
PI	=	π
POF	=	$\pi/4$
QOZ	=	Q(0,Z)
Q10	=	coefficient in Q(0,Z) = 1/8
Q11	=	coefficient in Q(1,Z) = 3/8
Q1Z	=	Q(1,Z)
Q20	=	coefficient in Q(0,Z) = $9 \times 25/(3!8^3)$
Q21	=	coefficient in Q(1,Z) = $(4 - 1)(4 - 9)(4 - 25)/(3!8^3)$
SZ	=	sin χ
TEST	=	magnitude of the term in the series
Z	=	Z
ZI	=	Z^2 or $1/Z$
ZI2	=	$1/Z^2$ or $\exp(-j\chi)$
ZK	=	$(-Z^2/4)^k/(k!)^2$ for series. Also temporary storage for asymptotic method
ZMS	=	$\|Z\|^2$ or temporary storage

CONSTANTS

31.41592654	=	$10.\pi$
36.	=	6^2
37.21	=	6.1^2

BESSEL

```
      SUBROUTINE BESSEL (Z,J0,J0P)
C
C     BESSEL EVALUATES THE ZERO-ORDER BESSEL FUNCTION AND ITS DERIVATIVE
C     FOR COMPLEX ARGUMENT Z.
C
      COMPLEX J0,J0P,P0Z,P1Z,Q0Z,Q1Z,Z,ZI,ZI2,ZK,FJ,CZ,SZ,J0X,J0PX
      DIMENSION M(101), A1(25), A2(25), FJX(2)
      EQUIVALENCE (FJ,FJX)
      DATA PI,C3,P10,P20,Q10,Q20/3.141592654,.7978845608,.0703125,.11215
     120996,.125,.0732421875/
      DATA P11,P21,Q11,Q21/.1171875,.1441955566,.375,.1025390625/
      DATA POF,INIT/.7853981635,0/,FJX/0.,1./
      IF (INIT.EQ.0) GO TO 5
    1 ZMS=Z*CONJG(Z)
      IF (ZMS.GT.1.E-12) GO TO 2
      J0=(1.,0.)
      J0P=-.5*Z
      RETURN
    2 IB=0
      IF (ZMS.GT.37.21) GO TO 4
      IF (ZMS.GT.36.) IB=1
C     SERIES EXPANSION
      IZ=1.+ZMS
      MIZ=M(IZ)
      J0=(1.,0.)
      J0P=J0
      ZK=J0
      ZI=Z*Z
      DO 3 K=1,MIZ
      ZK=ZK*A1(K)*ZI
      J0=J0+ZK
    3 J0P=J0P+A2(K)*ZK
      J0P=-.5*Z*J0P
      IF (IB.EQ.0) RETURN
      J0X=J0
      J0PX=J0P
C     ASYMPTOTIC EXPANSION
    4 ZI=1./Z
      ZI2=ZI*ZI
      P0Z=1.+(P20*ZI2-P10)*ZI2
      P1Z=1.+(P11-P21*ZI2)*ZI2
      Q0Z=(Q20*ZI2-Q10)*ZI
      Q1Z=(Q11-Q21*ZI2)*ZI
      ZK=CEXP(FJ*(Z-POF))
      ZI2=1./ZK
      CZ=.5*(ZK+ZI2)
      SZ=FJ*.5*(ZI2-ZK)
      ZK=C3*CSQRT(ZI)
      J0=ZK*(P0Z*CZ-Q0Z*SZ)
      J0P=-ZK*(P1Z*SZ+Q1Z*CZ)
      IF (IB.EQ.0) RETURN
      ZMS=COS((SQRT(ZMS)-6.)*31.41592654)
      J0=.5*(J0X*(1.+ZMS)+J0*(1.-ZMS))
      J0P=.5*(J0PX*(1.+ZMS)+J0P*(1.-ZMS))
      RETURN
C     INITIALIZATION OF CONSTANTS
    5 DO 6 K=1,25
      A1(K)=-.25/(K*K)
    6 A2(K)=1./(K+1.)
      DO 8 I=1,101
      TEST=1.
      DO 7 K=1,24
      INIT=K
      TEST=-TEST*I*A1(K)
```

```
      IF (TEST.LT.1.E-6) GO TO 8
7     CONTINUE
8     M(I)=INIT
      GO TO 1
      END
```

EVLUA

PURPOSE

To control the evaluation of the Sommerfeld integrals.

METHOD

The integration contour of either Figures 13, 14 or 15 of Part I is used depending on the values of ρ, $Z + Z'$ and k_1. Figures 13, 14, and 15 should be inverted, however, since they are for a time dependence of $\exp(j\omega t)$ and the coding for the Sommerfeld integrals is for $\exp(-j\omega t)$. Thus the contours and branch cuts in EVALUA are the conjugate of those shown. The conjugate of the results is taken at the end of EVLUA to conform to the NEC time dependence of $\exp(j\omega t)$.

The code from EV 19 to EV 34 evaluates the Bessel function form of the Sommerfeld integrals using the contour of Figure 13 of Part I. ROM1 is called to integrate from $\lambda = 0$ to $(p - jp)$ and GSHANK is called for the path from $(p - jp)$ to infinity where p^{-1} is the maximum of ρ and $Z + Z'$ (p = DEL). If p is greater than $100.|k_1|$ then ROM1 is called for the interval 0 to $(p_1 - jp_1)$ where $p_1 = 10|k_1|$. This is done to avoid exceeding the limit by which ROM1 can cut the interval width. Larger steps can then be used from $(p_1 - jp_1)$ to $(p - jp)$ since $\gamma_1 \approx \gamma_2 \approx \lambda$.

The code from EV 39 to EV 86 evaluates the Hankel function form of the integrals using either the contour of Figure 14 or 15. At EV 50 SUM is the negative of the integral from a^* to c^*. GSHANK is then called to integrate from a^* to $-\infty$. The decision whether to use the contour of Figures 14 or 15 is made from EV 58 to EV 64. Figure 15 is used if the real part of $\rho(k_1 - k_2)$ exceeds $2k_2$ and

$$\frac{-u}{|v|} > \frac{4\rho}{Z+Z'}$$

where $u + jv = [-(Z + Z') + j\rho][d^* - c^*]$ is the argument of the exponential function approximating the Sommerfeld integrand for large λ with $\lambda = d^* - c^*$. The left side of the inequality is proportional to the decay per cycle along the c to d path and $\rho/(Z + Z')$ is the same for the vertical path. This condition was chosen arbitrarily but gives some indication of when the contour of Figure 16 may be advantageous.

For the contour of Figure 15 GSHANK is called to integrate from e* to infinity. ROM1 is then called from e* to f*. The sign of the contribution from other parts of the path is switched since they were integrated in reverse direction. Finally, GSHANK is called for the paths from c* to infinity and f* to infinity.

For the contour of Figure 14 (GS 79 to GS 86) GSHANK is called to integrate from c* to d* and on to infinity. The increment changes from DELTA to DELTA2 if d* is reached before the integral converges.

From EV 89 to EV 92 the integrals are combined to form the field components and the conjugates are taken.

SYMBOL DICTIONARY

A	= start of integration interval
ANS	= temporary storage
B	= end of integration interval
BK	= break point (d*) in path for GSHANK
CK1	= k_1
CK1SQ	= k_1^2
CK2	= k_2
CK2SQ	= k_2^2
CP1	= a*
CP2	= b*
CP3	= c*
DEL	= p
DELTA	= increment along path
DELTA2	= alternate increment
EPH	= (see SOMNEC)
ERH	= (see SOMNEC)
ERV	= (see SOMNEC)
EZV	= (see SOMNEC)
JH	= 0 for Bessel function form, 1 for Hankel function form
PTP	= 0.2π
RHØ	= ρ

RMIS	=	temporary storage
SLOPE	=	slope of paths to infinity
SUM	=	temporary storage
TKMAG	=	$100\lvert k_1\rvert$
ZPH	=	$Z + Z'$

```
      SUBROUTINE EVLUA (ERV,EZV,ERH,EPH)
C
C     EVALUA CONTROLS THE INTEGRATION CONTOUR IN THE COMPLEX LAMBDA
C     PLANE FOR EVALUATION OF THE SOMMERFELD INTEGRALS.
C
      COMPLEX ERV,EZV,ERH,EPH,A,B,CK1,CK1SQ,BK,SUM,DELTA,ANS,DELTA2,CP1,
     1CP2,CP3,CKSM,CT1,CT2,CT3
      COMMON /CNTOUR/ A,B
      COMMON /EVLCOM/ CKSM,CT1,CT2,CT3,CK1,CK1SQ,CK2,CK2SQ,TKMAG,TSMAG,C
     1K1R,ZPH,RHO,JH
      DIMENSION SUM(6), ANS(6)
      DATA PTP/.6283185308/
      DEL=ZPH
      IF (RHO.GT.DEL) DEL=RHO
      IF (ZPH.LT.2.*RHO) GO TO 4
C
C     BESSEL FUNCTION FORM OF SOMMERFELD INTEGRALS
C
      JH=0
      A=(0.,0.)
      DEL=1./DEL
      IF (DEL.LE.TKMAG) GO TO 2
      B=CMPLX(.1*TKMAG,-.1*TKMAG)
      CALL ROM1 (6,SUM,2)
      A=B
      B=CMPLX(DEL,-DEL)
      CALL ROM1 (6,ANS,2)
      DO 1 I=1,6
1     SUM(I)=SUM(I)+ANS(I)
      GO TO 3
2     B=CMPLX(DEL,-DEL)
      CALL ROM1 (6,SUM,2)
3     DELTA=PTP*DEL
      CALL GSHANK (B,DELTA,ANS,6,SUM,0,B,B)
      GO TO 10
C
C     HANKEL FUNCTION FORM OF SOMMERFELD INTEGRALS
C
4     JH=1
      CP1=CMPLX(0.,.4*CK2)
      CP2=CMPLX(.6*CK2,-.2*CK2)
      CP3=CMPLX(1.02*CK2,-.2*CK2)
      A=CP1
      B=CP2
      CALL ROM1 (6,SUM,2)
      A=CP2
      B=CP3
      CALL ROM1 (6,ANS,2)
      DO 5 I=1,6
5     SUM(I)=-(SUM(I)+ANS(I))
C     PATH FROM IMAGINARY AXIS TO -INFINITY
      SLOPE=1000.
      IF (ZPH.GT..001*RHO) SLOPE=RHO/ZPH
      DEL=PTP/DEL
      DELTA=CMPLX(-1.,SLOPE)*DEL/SQRT(1.+SLOPE*SLOPE)
      DELTA2=-CONJG(DELTA)
      CALL GSHANK (CP1,DELTA,ANS,6,SUM,0,BK,BK)
      RMIS=RHO*(REAL(CK1)-CK2)
      IF (RMIS.LT.2.*CK2) GO TO 8
      IF (RHO.LT.1.E-10) GO TO 8
      IF (ZPH.LT.1.E-10) GO TO 6
      BK=CMPLX(-ZPH,RHO)*(CK1-CP3)
      RMIS=-REAL(BK)/ABS(AIMAG(BK))
      IF(RMIS.GT.4.*RHO/ZPH)GO TO 8
```

```
C     INTEGRATE UP BETWEEN BRANCH CUTS, THEN TO + INFINITY            EV  65
6     CP1=CK1-(.1,.2)                                                 EV  66
      CP2=CP1+.2                                                      EV  67
      BK=CMPLX(0.,DEL)                                                EV  68
      CALL GSHANK (CP1,BK,SUM,6,ANS,0,BK,BK)                          EV  69
      A=CP1                                                           EV  70
      B=CP2                                                           EV  71
      CALL ROM1 (6,ANS,1)                                             EV  72
      DO 7 I=1,6                                                      EV  73
7     ANS(I)=ANS(I)-SUM(I)                                            EV  74
      CALL GSHANK (CP3,BK,SUM,6,ANS,0,BK,BK)                          EV  75
      CALL GSHANK (CP2,DELTA2,ANS,6,SUM,0,BK,BK)                      EV  76
      GO TO 10                                                        EV  77
C     INTEGRATE BELOW BRANCH POINTS, THEN TO + INFINITY               EV  78
8     DO 9 I=1,6                                                      EV  79
9     SUM(I)=-ANS(I)                                                  EV  80
      RMIS=REAL(CK1)*1.01                                             EV  81
      IF (CK2+1..GT.RMIS) RMIS=CK2+1.                                 EV  82
      BK=CMPLX(RMIS,.99*AIMAG(CK1))                                   EV  83
      DELTA=BK-CP3                                                    EV  84
      DELTA=DELTA*DEL/CABS(DELTA)                                     EV  85
      CALL GSHANK (CP3,DELTA,ANS,6,SUM,1,BK,DELTA2)                   EV  86
10    ANS(6)=ANS(6)*CK1                                               EV  87
C     CONJUGATE SINCE NEC USES EXP(+JWT)                              EV  88
      ERV=CONJG(CK1SQ*ANS(3))                                         EV  89
      EZV=CONJG(CK1SQ*(ANS(2)+CK2SQ*ANS(5)))                          EV  90
      ERH=CONJG(CK2SQ*(ANS(1)+ANS(6)))                                EV  91
      EPH=-CONJG(CK2SQ*(ANS(4)+ANS(6)))                               EV  92
      RETURN                                                          EV  93
      END                                                             EV  94-
```

GSHANK

PURPOSE

To apply the Shanks transformation (ref. 6) to accelerate the convergence of a semi-infinite integral.

METHOD

Six integrals (NANS = 6) are evaluated simultaneously in this routine. The integrals over semi-infinite sections of the contours (Figures 13, 14 and 15 of Part I) are evaluated by using the Romberg variable interval width integration method on subsections to obtain a converging sequence of partial sums

$$S_i = S_0 + \int_{A_0}^{A_0+i\Delta} f(\lambda)d\lambda \qquad i = 1, 2, \ldots$$

where A_0 is the start of the semi-infinite path, S_0 is the contribution from other parts of the contour and Δ is a complex increment with

$$|\Delta| = \text{minimum of} \begin{cases} 0.2\pi/\rho \\ 0.2\pi/(Z + Z') \end{cases}$$

$$\arg(\Delta) = \text{direction of integration path in } \lambda\text{-plane}$$

The Shanks interated first order transformation is applied to S_i to accelerate convergence. Starting with the sequence of M elements $Q_{i,0} = S_i$, i = 1, ... M the j^{th} iterated transform is the sequence of M - 2j elements

$$Q_{ij} = \frac{Q_{i-1,j-1}Q_{i+1,j-1} - Q^2_{i,\,j-1}}{Q_{i-1,j-1} + Q_{i+1,j-1} - 2Q_{i,\,j-1}}$$

$$= Q_{i-1,j-1} - \frac{(Q_{i,j-1} - Q_{i-1,j-1})^2}{Q_{i-1,j-1} + Q_{i+1,j-1} - 2Q_{i,j-1}}$$

$i = j + 1, \ldots M - j$
$j = 1, \ldots [(M - 1)/2].$

The second form for $Q_{i,j}$ is used since it suffers less numerical error as the sequence converges. Each iteration of the transform should produce a sequence that converges more rapidly to the limit of the original sequence.

In this subroutine the starting value S_0 comes in as SEED. With each pass through the loop over INT, starting at GS 21, two new values are added to the sequence by calling ROM1 to evaluate the integrals

$$S_{2N-1} = S_{2N-2} + \int_{A_0+(2N-2)\Delta}^{A_0+(2N-1)\Delta} f(\lambda)d\lambda$$

$$S_{2N} = S_{2N-1} + \int_{A_0+(2N-1)\Delta}^{A_0+(2N)\Delta} f(\lambda)d\lambda$$

where N = INT. The $(N - 1)^{th}$ interated Shanks transformation, consisting of the two elements $Q_{N,N-1}$ and $Q_{N+1,N-1}$, is then computed. At the end of each pass through the loop over INT the arrays Q1 and Q2 contain the last two elements in each sequence. For function I,

$$Q1(I,J) = Q_{2N-J,J-1}$$

$$Q2(I,J) = Q_{2N-J+1,J-1}, \quad J = 1, \ldots N.$$

For the path from c to infinity in Figure 14 of Part I the point d is a break point at which Δ may change. If d is reached before convergence the Shanks transformation is started over with the final value of S_i becoming S_0 for the new sequence.

Convergence is tested from GS 78 to GS 89 by comparing the last two values in the transformed sequences. Although the last sequence, consisting of two elements, should have the highest convergence the last four sequences are tested to avoid a false indication of convergence. The relative difference is computed for each of the six functions and compared with CRIT. If convergence does not occur by INT = MAXH a message is printed and the average of the two values in the last sequence is used for each integral. In computing the relative difference for each function the denominator is not allowed to be less than 10^{-3} times the magnitude of the largest of the six functions to avoid convergence problems when one function goes to zero.

SYMBOL DICTIONARY

A	= beginning of integration subinterval
A1	= new value for Q1 array
A2	= new value for Q2 array
AA	= temporary storage
AMG	= approximate magnitude of function
ANS1	= S_i for i odd
ANS2	= S_i for i even
AS1	= S_i for i odd
AS2	= S_i for i even
B	= end of integration subinterval
BK	= break point in integration contour
CRIT	= limit for relative error in convergence test
DEL	= Δ
DELA	= Δ before break point
DELB	= Δ after break point
DEN	= approximate magnitude of the largest of the six functions (GS 76)
DENM	= minimum denominator for relative error test
IBK	= 1 if path contains break point
IBX	= 0 if path contains break point and it has not been passed
INT	= N

INX = INT

JM = J - 1

MAXH = maximum for index J in Q1 and Q2

NANS = number of functions (6)

Q1, Q2 = (see description of method)

RBK = real part of BK

SEED = S_0

START = A_0

SUM = increment to integral

```
      SUBROUTINE GSHANK (START,DELA,SUM,NANS,SEED,IBK,BK,DELB)
C
C     GSHANK INTEGRATES THE 6 SOMMERFELD INTEGRALS FROM START TO
C     INFINITY (UNTIL CONVERGENCE) IN LAMBDA.  AT THE BREAK POINT, BK,
C     THE STEP INCREMENT MAY BE CHANGED FROM DELA TO DELB.  SHANK S
C     ALGORITHM TO ACCELERATE CONVERGENCE OF A SLOWLY CONVERGING SERIES
C     IS USED
C
      COMPLEX START,DELA,SUM,SEED,BK,DELB,A,B,Q1,Q2,ANS1,ANS2,A1,A2,AS1,
     1AS2,DEL,AA
      COMMON /CNTOUR/ A,B
      DIMENSION Q1(6,20), Q2(6,20), ANS1(6), ANS2(6), SUM(6), SEED(6)
      DATA CRIT/1.E-4/,MAXH/20/
      RBK=REAL(BK)
      DEL=DELA
      IBX=0
      IF (IBK.EQ.0) IBX=1
      DO 1 I=1,NANS
1     ANS2(I)=SEED(I)
      B=START
2     DO 20 INT=1,MAXH
      INX=INT
      A=B
      B=B+DEL
      IF (IBX.EQ.0.AND.REAL(B).GE.RBK) GO TO 5
      CALL ROM1 (NANS,SUM,2)
      DO 3 I=1,NANS
3     ANS1(I)=ANS2(I)+SUM(I)
      A=B
      B=B+DEL
      IF (IBX.EQ.0.AND.REAL(B).GE.RBK) GO TO 6
      CALL ROM1 (NANS,SUM,2)
      DO 4 I=1,NANS
4     ANS2(I)=ANS1(I)+SUM(I)
      GO TO 11
C     HIT BREAK POINT.  RESET SEED AND START OVER.
5     IBX=1
      GO TO 7
6     IBX=2
7     B=BK
      DEL=DELB
      CALL ROM1 (NANS,SUM,2)
      IF (IBX.EQ.2) GO TO 9
      DO 8 I=1,NANS
8     ANS2(I)=ANS2(I)+SUM(I)
      GO TO 2
9     DO 10 I=1,NANS
10    ANS2(I)=ANS1(I)+SUM(I)
      GO TO 2
11    DEN=0.
      DO 18 I=1,NANS
      AS1=ANS1(I)
      AS2=ANS2(I)
      IF (INT.LT.2) GO TO 17
      DO 16 J=2,INT
      JM=J-1
      AA=Q2(I,JM)
      A1=Q1(I,JM)+AS1-2.*AA
      IF (REAL(A1).EQ.0..AND.AIMAG(A1).EQ.0.) GO TO 12
      A2=AA-Q1(I,JM)
      A1=Q1(I,JM)-A2*A2/A1
      GO TO 13
12    A1=Q1(I,JM)
13    A2=AA+AS2-2.*AS1
```

```
      IF (REAL(A2).EQ.0..AND.AIMAG(A2).EQ.0.) GO TO 14
      A2=AA-(AS1-AA)*(AS1-AA)/A2
      GO TO 15
   14 A2=AA
   15 Q1(I,JM)=AS1
      Q2(I,JM)=AS2
      AS1=A1
   16 AS2=A2
   17 Q1(I,INT)=AS1
      Q2(I,INT)=AS2
      AMG=ABS(REAL(AS2))+ABS(AIMAG(AS2))
      IF (AMG.GT.DEN) DEN=AMG
   18 CONTINUE
      DENM=1.E-3*DEN*CRIT
      JM=INT-3
      IF (JM.LT.1) JM=1
      DO 19 J=JM,INT
      DO 19 I=1,NANS
      A1=Q2(I,J)
      DEN=(ABS(REAL(A1))+ABS(AIMAG(A1)))*CRIT
      IF (DEN.LT.DENM) DEN=DENM
      A1=Q1(I,J)-A1
      AMG=ABS(REAL(A1))+ABS(AIMAG(A1))
      IF (AMG.GT.DEN) GO TO 20
   19 CONTINUE
      GO TO 22
   20 CONTINUE
      PRINT 24
      DO 21 I=1,NANS
   21 PRINT 25, Q1(I,INX),Q2(I,INX)
   22 DO 23 I=1,NANS
   23 SUM(I)=.5*(Q1(I,INX)+Q2(I,INX))
      RETURN
C
   24 FORMAT (46H **** NO CONVERGENCE IN SUBROUTINE GSHANK ****)
   25 FORMAT (10E12.5)
      END
```

HANKEL

PURPOSE

To compute the Hankel function of the first kind, zeroth order, and its derivative for a complex argument.

METHOD

For argument magnitudes less than a limit Z_s the functions are evaluated by the ascending series and for larger magnitudes by Hankel's asymptotic expansion (ref. 5). The series are

$$Y_0(Z) = \frac{2}{\pi} \ln(Z/2) J_0(Z) - \frac{2}{\pi} \sum_{k=0}^{\infty} \psi(k+1) \frac{(-Z^2/4)^k}{(k!)^2}$$

$$Y_0'(Z) = \frac{2}{\pi Z} + \frac{2}{\pi} \ln(Z/2) J_0'(Z) + \frac{Z}{2\pi} \sum_{k=0}^{\infty} [\psi(k+1) + \psi(k+2)] \frac{(-Z^2/4)^k}{k!(k+1)!}$$

where $\psi(k+1) = -\gamma + \sum_{j=1}^{k} j^{-1}$

$\psi(1) = -\gamma$,

γ = Euler's constant = 0.5772156649

The Hankel functions are

$$H_0^{(1)}(Z) = J_0(Z) + j\, Y_0(Z)$$

$$H_0^{(1)'}(Z) = J_0'(Z) + j\, Y_0'(Z)$$

The series for $J_0(Z)$ and $J_0'(Z)$ are given in the description of subroutine BESSEL. The number of terms used with an argument Z is M(IZ) where $IZ = 1. + |Z|^2$.

The array M is filled for IZ from 1 to 101 on the first call to HANKEL by determining the value of k at which the term in the series of Y_0 is less than 10^{-6}.

HANKEL

When $|Z|$ is greater than Z_s Hankel's asymptotic expansions are used with two or three terms. These are

$$H_\nu^{(1)}(Z) = \sqrt{\frac{2}{\pi Z}}\,[P(\nu,Z) + jQ(\nu,Z)]e^{j\chi}$$

$$\chi = Z - (\frac{1}{2}\nu + \frac{1}{4})\pi$$

$P(\nu,Z)$ and $Q(\nu,Z)$ are given in the description of subroutine BESSEL.

When $Z_s < |Z| < Z_s + \Delta$ both the series and asymptotic forms are evaluated and are combined as in BESSEL to eliminate any discontinuity. In HANKEL Z_s is 4 and Δ is 0.1.

SYMBOL DICTIONARY

A1	=	$-1./(4k^2)$
A2	=	$1./(k+1)$
A3	=	$2\psi(k+1)$
A4	=	$[\psi(k+1) + \psi(k+2)]/(k+1)$
C1	=	$[\psi(1) + \psi(2)]/(2\pi)$
C2	=	$2\gamma/\pi.$
C3	=	$\sqrt{2/\pi}$
CLOGZ	=	$\ln(Z)$
FJ	=	$\sqrt{-1}$
FJX	=	FJ
GAMMA	=	γ
HO	=	$H_0^{(1)}(Z)$
HOP	=	$H_0^{(1)'}(Z)$
IB	=	1 to indicate that both the series and asymptotic forms will be evaluated and combined
INIT	=	flag to indicate that initialization of constants has been completed
IZ	=	$1. + \|Z\|^2$
JO	=	$J_0(Z)$
JOP	=	$J_0'(Z)$

K = summation index k, summed from 1 to limit

M = array of upper limits for k

MIZ = upper limit for k

POZ, P10, P11, P1Z, P20, P21: see BESSEL

PI = π

POF = $\pi/4$

PSI = ψ

QOZ, Q10, Q11, Q1Z, Q20, Q21: see BESSEL

TEST = magnitude of term in the series

YO = $Y_0(Z)$

YOP = $Y_0'(Z)$

Z = Z

ZI = Z^2 or $1/Z$

ZI2 = $1/Z^2$

ZK = $(-Z^2/4)^k/(k!)^2$; also temporary storage

ZMS = $|Z^2|$ or temporary storage

CONSTANTS

16.	$= 4^2$
16.81	$= 4.1^2$
31.41592654	$= 10.\pi$

```
      SUBROUTINE HANKEL (Z,HO,HOP)
C
C     HANKEL EVALUATES HANKEL FUNCTION OF THE FIRST KIND, ORDER ZERO,
C     AND ITS DERIVATIVE FOR COMPLEX ARGUMENT Z.
C
      COMPLEX CLOGZ,HO,HOP,JO,JOP,POZ,P1Z,QOZ,Q1Z,YO,YOP,Z,ZI,ZI2,ZK,FJ
      DIMENSION M(101), A1(25), A2(25), A3(25), A4(25), FJX(2)
      EQUIVALENCE (FJ,FJX)
      DATA PI,GAMMA,C1,C2,C3,P10,P20/3.141592654,.5772156649,-.024578509
     15,.3674669052,.7978845608,.0703125,.1121520996/
      DATA Q10,Q20,P11,P21,Q11,Q21/.125,.0732421875,.1171875,.1441955566
     1,.375,.1025390625/
      DATA POF,INIT/.7853981635,0/,FJX/0.,1./
      IF (INIT.EQ.0) GO TO 5
    1 ZMS=Z*CONJG(Z)
      IF (ZMS.NE.0.) GO TO 2
      PRINT 9
      STOP
    2 IB=0
      IF (ZMS.GT.16.81) GO TO 4
      IF (ZMS.GT.16.) IB=1
C     SERIES EXPANSION
      IZ=1.+ZMS
      MIZ=M(IZ)
      JO=(1.,0.)
      JOP=JO
      YO=(0.,0.)
      YOP=YO
      ZK=JO
      ZI=Z*Z
      DO 3 K=1,MIZ
      ZK=ZK*A1(K)*ZI
      JO=JO+ZK
      JOP=JOP+A2(K)*ZK
      YO=YO+A3(K)*ZK
    3 YOP=YOP+A4(K)*ZK
      JOP=-.5*Z*JOP
      CLOGZ=CLOG(.5*Z)
      YO=(2.*JO*CLOGZ-YO)/PI+C2
      YOP=(2./Z+2.*JOP*CLOGZ+.5*YOP*Z)/PI+C1*Z
      HO=JO+FJ*YO
      HOP=JOP+FJ*YOP
      IF (IB.EQ.0) RETURN
      YO=HO
      YOP=HOP
C     ASYMPTOTIC EXPANSION
    4 ZI=1./Z
      ZI2=ZI*ZI
      POZ=1.+(P20*ZI2-P10)*ZI2
      P1Z=1.+(P11-P21*ZI2)*ZI2
      QOZ=(Q20*ZI2-Q10)*ZI
      Q1Z=(Q11-Q21*ZI2)*ZI
      ZK=CEXP(FJ*(Z-POF))*CSQRT(ZI)*C3
      HO=ZK*(POZ+FJ*QOZ)
      HOP=FJ*ZK*(P1Z+FJ*Q1Z)
      IF (IB.EQ.0) RETURN
      ZMS=COS((SQRT(ZMS)-4.)*31.41592654)
      HO=.5*(YO*(1.+ZMS)+HO*(1.-ZMS))
      HOP=.5*(YOP*(1.+ZMS)+HOP*(1.-ZMS))
      RETURN
C     INITIALIZATION OF CONSTANTS
    5 PSI=-GAMMA
      DO 6 K=1,25
      A1(K)=-.25/(K*K)
```

```
      A2(K)=1./(K+1.)                                          HA  65
      PSI=PSI+1./K                                             HA  66
      A3(K)=PSI+PSI                                            HA  67
    6 A4(K)=(PSI+PSI+1./(K+1.))/(K+1.)                         HA  68
      DO 8 I=1,101                                             HA  69
      TEST=1.                                                  HA  70
      DO 7 K=1,24                                              HA  71
      INIT=K                                                   HA  72
      TEST=-TEST*I*A1(K)                                       HA  73
      IF (TEST*A3(K).LT.1.E-6) GO TO 8                         HA  74
    7 CONTINUE                                                 HA  75
    8 M(I)=INIT                                                HA  76
      GO TO 1                                                  HA  77
C                                                              HA  78
    9 FORMAT (34H ERROR - HANKEL NOT VALID FOR Z=0.)           HA  79
      END                                                      HA  80-
```

LAMBDA

PURPOSE

To compute the complex value of λ from the real integration parameter in ROM1.

METHOD

For integration along a straight path between the points a and b in the λ plane, λ and $d\lambda$ are

$$\lambda = a + (b - a)t$$

$$d\lambda = (b - a)dt$$

SYMBOL DICTIONARY

A	= a
B	= b
DXLAM	= b - a
T	= t
XLAM	= λ

```
      SUBROUTINE LAMBDA (T,XLAM,DXLAM)                                  LA   1
C                                                                       LA   2
C     COMPUTE INTEGRATION PARAMETER XLAM=LAMBDA FROM PARAMETER T.      LA   3
C                                                                       LA   4
      COMPLEX A,B,XLAM,DXLAM                                            LA   5
      COMMON /CNTOUR/ A,B                                               LA   6
      DXLAM=B-A                                                         LA   7
      XLAM=A+DXLAM*T                                                    LA   8
      RETURN                                                            LA   9
      END                                                               LA  10-
```

ROM1

PURPOSE

To integrate the Sommerfeld integrands between two points in λ by the method of variable interval-width Romberg integration.

METHOD

A and B in common block /CNTOUR/ are the ends of the integration path and are set before ROM1 is called. The integration parameter Z in ROM1 starts at zero and ends at one. The corresponding value of λ is determined by subroutine LAMBDA as

$$\lambda = A + (B - A)Z$$

Subroutine SAOA returns six integrand values which are handled simultaneously in loops throughout the code. The Romberg variable interval-width integration method will not be described in detail since it is the same as that used in subroutine INTX in the main NEC program. The convergence test in ROM1 requires that all six components satisfy the relative error tests simultaneously.

```
      SUBROUTINE ROM1 (N,SUM,NX)
C
C     ROM1 INTEGRATES THE 6 SOMMERFELD INTEGRALS FROM A TO B IN LAMBDA.
C     THE METHOD OF VARIABLE INTERVAL WIDTH ROMBERG INTEGRATION IS USED.
C
      COMPLEX A,B,SUM,G1,G2,G3,G4,G5,T00,T01,T10,T02,T11,T20
      COMMON /CNTOUR/ A,B
      DIMENSION SUM(6), G1(6), G2(6), G3(6), G4(6), G5(6), T01(6), T10(6
     1), T20(6)
      DATA NM,NTS,RX/131072,4,1.E-4/
      LSTEP=0
      Z=0.
      ZE=1.
      S=1.
      EP=S/(1.E4*NM)
      ZEND=ZE-EP
      DO 1 I=1,N
1     SUM(I)=(0.,0.)
      NS=NX
      NT=0
      CALL SAOA (Z,G1)
2     DZ=S/NS
      IF (Z+DZ.LE.ZE) GO TO 3
      DZ=ZE-Z
      IF (DZ.LE.EP) GO TO 17
3     DZOT=DZ*.5
      CALL SAOA (Z+DZOT,G3)
      CALL SAOA (Z+DZ,G5)
4     NOGO=0
      DO 5 I=1,N
      T00=(G1(I)+G5(I))*DZOT
      T01(I)=(T00+DZ*G3(I))*.5
      T10(I)=(4.*T01(I)-T00)/3.
C     TEST CONVERGENCE OF 3 POINT ROMBERG RESULT
      CALL TEST (REAL(T01(I)),REAL(T10(I)),TR,AIMAG(T01(I)),AIMAG(T10(I)
     1),TI,0.)
      IF (TR.GT.RX.OR.TI.GT.RX) NOGO=1
5     CONTINUE
      IF (NOGO.NE.0) GO TO 7
      DO 6 I=1,N
6     SUM(I)=SUM(I)+T10(I)
      NT=NT+2
      GO TO 11
7     CALL SAOA (Z+DZ*.25,G2)
      CALL SAOA (Z+DZ*.75,G4)
      NOGO=0
      DO 8 I=1,N
      T02=(T01(I)+DZOT*(G2(I)+G4(I)))*.5
      T11=(4.*T02-T01(I))/3.
      T20(I)=(16.*T11-T10(I))/15.
C     TEST CONVERGENCE OF 5 POINT ROMBERG RESULT
      CALL TEST (REAL(T11),REAL(T20(I)),TR,AIMAG(T11),AIMAG(T20(I)),TI,0
     1.)
      IF (TR.GT.RX.OR.TI.GT.RX) NOGO=1
8     CONTINUE
      IF (NOGO.NE.0) GO TO 13
9     DO 10 I=1,N
10    SUM(I)=SUM(I)+T20(I)
      NT=NT+1
11    Z=Z+DZ
      IF (Z.GT.ZEND) GO TO 17
      DO 12 I=1,N
12    G1(I)=G5(I)
      IF (NT.LT.NTS.OR.NS.LE.NX) GO TO 2
```

```
      NS=NS/2
      NT=1
      GO TO 2
   13 NT=0
      IF (NS.LT.NM) GO TO 15
      IF (LSTEP.EQ.1) GO TO 9
      LSTEP=1
      CALL LAMBDA (Z,T00,T11)
      PRINT 18, T00
      PRINT 19, Z,DZ,A,B
      DO 14 I=1,N
   14 PRINT 19, G1(I),G2(I),G3(I),G4(I),G5(I)
      GO TO 9
   15 NS=NS*2
      DZ=S/NS
      DZOT=DZ*.5
      DO 16 I=1,N
      G5(I)=G3(I)
   16 G3(I)=G2(I)
      GO TO 4
   17 CONTINUE
      RETURN
C
   18 FORMAT (38H ROM1 -- STEP SIZE LIMITED AT LAMBDA =,2E12.5)
   19 FORMAT (10E12.5)
      END
```

SAOA

PURPOSE

To compute the integrands for the Sommerfeld integrals.

METHOD

The input to SAOA is the integration parameter T and constants in common block /EVLCOM/. The integration variable λ corresponding to T is obtained by calling subroutine LAMBDA. The values returned in array ANS are

$$ANS(1) = D_2 H_0^{(1)''}(\lambda\rho) e^{-\gamma_2(Z+Z')} \lambda^3 d\lambda/dT$$

$$ANS(2) = D_2 \gamma_2^2 H_0^{(1)}(\lambda\rho) e^{-\gamma_2(Z+Z')} \lambda\, d\lambda/dT$$

$$ANS(3) = -D_2 \gamma_2 H_0^{(1)'}(\lambda\rho) e^{-\gamma_2(Z+Z')} \lambda^2 d\lambda/dT$$

$$ANS(4) = \rho^{-1} D_2 H_0^{(1)'}(\lambda\rho) e^{-\gamma_2(Z+Z')} \lambda^2 d\lambda/dT$$

$$ANS(5) = D_2 H_0^{(1)}(\lambda\rho) e^{-\gamma_2(Z+Z')} \lambda d\lambda/dT$$

$$ANS(6) = k_1^{-1} D_1 H_0^{(1)}(\lambda\rho) e^{-\gamma_2(Z+Z')} \lambda d\lambda/dT$$

where $D_1 = \dfrac{1}{\gamma_1+\gamma_2} - \dfrac{k_2^2}{\gamma_2(k_1^2+k_2^2)}$

$$D_2 = \frac{1}{k_1^2\gamma_2+k_2^2\gamma_1} - \frac{1}{\gamma_2(k_1^2+k_2^2)} = \frac{k_2^2(\gamma_2-\gamma_1)}{\gamma_2(k_1^2+k_2^2)(k_1^2\gamma_2+k_2^2\gamma_1)}$$

$$\gamma_1 = [\lambda^2 - k_1^2]^{1/2}$$

$$\gamma_2 = [\lambda^2 - k_2^2]^{1/2}$$

$$k_1 = k_2(\epsilon_1 - j\sigma_1/\omega\epsilon_0)^{1/2}$$

$$k_2 = \omega\sqrt{\mu_0\epsilon_0}$$

The integrands given above are computed when JH >0. When JH ≤ 0, $H_0^{(1)}(\lambda\rho)$ is replaced by $2J_0(\lambda\rho)$. The functions γ_1 and γ_2 are computed from SA 24 to SA 29 so that the branch cuts are vertical. This is not necessary from SA 17 to SA 20 since for the Bessel function form the integration contour is confined to a different quadrant than the branch cuts.

To avoid loss of accuracy due to cancellation when λ is large, D_2 is computed from the approximation for $\gamma_2 - \gamma_1$:

$$\gamma_2 - \gamma_1 \approx \pm\left[\frac{1}{2}\,\frac{k_1^2-k_2^2}{\lambda} + \frac{1}{8}\,\frac{k_1^4-k_2^4}{\lambda^3} + \frac{1}{16}\,\frac{k_1^6-k_2^6}{\lambda^5}\right]$$

when $|\lambda|^2 \geq 100.\,|k_1|^2$.

The sign is:

$$\begin{aligned} &- \text{ for } \lambda_R < k_{2_R},\ \lambda_I \geq 0\\ &- \text{ for } \lambda_R < -k_{1_R},\ \lambda_I < 0\\ &+ \text{ for } \lambda_R > k_{1_R},\ \lambda_I \geq 0\\ &+ \text{ for } \lambda_R > -k_{2_R},\ \lambda_I < 0. \end{aligned}$$

There is no cancellation and this approximation is not valid when

$$\begin{aligned} &k_{2_R} \leq \lambda_R \leq k_{1_R},\ \lambda_I \geq 0\\ \text{or}\quad &-k_{1_r} \leq \lambda_R \leq -k_{2_R},\ \lambda_I < 0. \end{aligned}$$

D_1 and D_2 are computed from SA 30 to SA 44.

SYMBOL DICTIONARY

ANS	= integrand values
BO	= $2J_{\rho}(\lambda\rho)$ or $H_0^{(1)}(\lambda\rho)$
BOP	= $2J_0(\lambda\rho)/\rho$ or $H_0^{(1)'}(\lambda\rho)/\rho$
CGAM1	= γ_1
CGAM2	= γ_2
CK1	= k_1
CK1R	= real part of k_1
CK1SQ	= k_1^2
CK2	= k_2
CK2SQ	= k_2^2
CKSM	= $k_2^2/(k_1^2+k_2^2)$
COM	= $\exp\,[-\gamma_2(Z+Z')]\lambda d\lambda/dT$ at SA 45
CT1	= $(k_1^2-k_2^2)/2$
CT2	= $(k_1^4-k_2^4)/8$
CT3	= $(k_1^6-k_2^6)/16$
DEN1	= D_1
DEN2	= D_2
DGAM	= $\gamma_2 - \gamma_1$
DXL	= $d\lambda/dT$
JH	= flag to select Bessel or Hankel function form
RHO	= ρ
SIGN	= sign in approximation for $\gamma_2 - \gamma_1$
T	= integration parameter
TKMAG	= $100.\lvert k_1\rvert$
TSMAG	= $100.\lvert k_1\rvert^2$
XL	= λ
XLR	= real part of λ
ZPH	= $Z + Z'$

```
      SUBROUTINE SAOA (T,ANS)                                           SA   1
C                                                                       SA   2
C     SAOA COMPUTES THE INTEGRAND FOR EACH OF THE 6                     SA   3
C     SOMMERFELD INTEGRALS FOR SOURCE AND OBSERVER ABOVE GROUND         SA   4
C                                                                       SA   5
      COMPLEX ANS,XL,DXL,CGAM1,CGAM2,B0,B0P,COM,CK1,CK1SQ,CKSM,CT1,CT2,C SA   6
     1T3,DGAM,DEN1,DEN2                                                 SA   7
      COMMON /EVLCOM/ CKSM,CT1,CT2,CT3,CK1,CK1SQ,CK2,CK2SQ,TKMAG,TSMAG,C SA   8
     1K1R,ZPH,RHO,JH                                                    SA   9
      DIMENSION ANS(6)                                                  SA  10
      CALL LAMBDA (T,XL,DXL)                                            SA  11
      IF (JH.GT.0) GO TO 1                                              SA  12
C     BESSEL FUNCTION FORM                                              SA  13
      CALL BESSEL (XL*RHO,B0,B0P)                                       SA  14
      B0=2.*B0                                                          SA  15
      B0P=2.*B0P                                                        SA  16
      CGAM1=CSQRT(XL*XL-CK1SQ)                                          SA  17
      CGAM2=CSQRT(XL*XL-CK2SQ)                                          SA  18
      IF (REAL(CGAM1).EQ.0.) CGAM1=CMPLX(0.,-ABS(AIMAG(CGAM1)))         SA  19
      IF (REAL(CGAM2).EQ.0.) CGAM2=CMPLX(0.,-ABS(AIMAG(CGAM2)))         SA  20
      GO TO 2                                                           SA  21
C     HANKEL FUNCTION FORM                                              SA  22
1     CALL HANKEL (XL*RHO,B0,B0P)                                       SA  23
      COM=XL-CK1                                                        SA  24
      CGAM1=CSQRT(XL+CK1)*CSQRT(COM)                                    SA  25
      IF (REAL(COM).LT.0..AND.AIMAG(COM).GE.0.) CGAM1=-CGAM1            SA  26
      COM=XL-CK2                                                        SA  27
      CGAM2=CSQRT(XL+CK2)*CSQRT(COM)                                    SA  28
      IF (REAL(COM).LT.0..AND.AIMAG(COM).GE.0.) CGAM2=-CGAM2            SA  29
2     XLR=XL*CONJG(XL)                                                  SA  30
      IF (XLR.LT.TSMAG) GO TO 3                                         SA  31
      IF (AIMAG(XL).LT.0.) GO TO 4                                      SA  32
      XLR=REAL(XL)                                                      SA  33
      IF (XLR.LT.CK2) GO TO 5                                           SA  34
      IF (XLR.GT.CK1R) GO TO 4                                          SA  35
3     DGAM=CGAM2-CGAM1                                                  SA  36
      GO TO 7                                                           SA  37
4     SIGN=1.                                                           SA  38
      GO TO 6                                                           SA  39
5     SIGN=-1.                                                          SA  40
6     DGAM=1./(XL*XL)                                                   SA  41
      DGAM=SIGN*((CT3*DGAM+CT2)*DGAM+CT1)/XL                            SA  42
7     DEN2=CKSM*DGAM/(CGAM2*(CK1SQ*CGAM2+CK2SQ*CGAM1))                  SA  43
      DEN1=1./(CGAM1+CGAM2)-CKSM/CGAM2                                  SA  44
      COM=DXL*XL*CEXP(-CGAM2*ZPH)                                       SA  45
      ANS(6)=COM*B0*DEN1/CK1                                            SA  46
      COM=COM*DEN2                                                      SA  47
      IF (RHO.EQ.0.) GO TO 8                                            SA  48
      B0P=B0P/RHO                                                       SA  49
      ANS(1)=-COM*XL*(B0P+B0*XL)                                        SA  50
      ANS(4)=COM*XL*B0P                                                 SA  51
      GO TO 9                                                           SA  52
8     ANS(1)=-COM*XL*XL*.5                                              SA  53
      ANS(4)=ANS(1)                                                     SA  54
9     ANS(2)=COM*CGAM2*CGAM2*B0                                         SA  55
      ANS(3)=-ANS(4)*CGAM2*RHO                                          SA  56
      ANS(5)=COM*B0                                                     SA  57
      RETURN                                                            SA  58
      END                                                               SA  59-
```

SECOND - see SECOND in main NEC program.

TEST - see TEST in main NEC program.

2. COMMON BLOCKS IN SOMNEC

COMMON/CNTOUR/ A, B

Routines Using /CNTOUR/

EVLUA, GSHANK, LAMBDA, ROM1

Parameters

A = start of integration interval

B = end of integration interval

A and B are used by subroutine LAMBDA to compute the complex value of λ from the real parameter supplied by ROM1.

COMMON/EVLCOM/ CKSM, CT1, CT2, CT3, CK1, CK1SQ, CK2, CK2SQ, TKMAG, TSMAG, CK1R, ZPH, RHO, JH

Routines Using /EVLCOM/

SOMNEC, EVLUA, SAOA

Parameters

See symbol dictionaries for subroutines

COMMON/GGRID/ AR1 (11, 10, 4), AR2 (17, 5, 4), AR3 (9, 8, 4), EPSCF, DXA(3), DYA(3), XSA(3), YSA(3), NXA(3), NYA(3)

Routines Using /GGRID/

SOMNEC (main program)

Parameters

AR1 = array for grid 1 (see Figure 12, Part I)

AR2 = array for grid 2

AR3 = array for grid 3

EPSCF = ε_c

For grid i, ARi(j, k, m) is the value of I_ρ^V, I_z^V, I_ρ^H, or I_ϕ^H for M = 1, ... 4 respectively at the point

$$R_1/\lambda = S_i + (j-1)\Delta R_i \qquad j = 1, \ldots N_i$$
$$\theta = T_i + (k-1)\Delta\theta_i \qquad k = 1, \ldots M_i$$

where S_i = XSA(i)

ΔR_i = DXA(i)

N_i = NXA(i)

T_i = YSA(i)

$\Delta\theta_i$ = DYA(i)

M_i = NYA(i)

XSA and DXA are in units of wavelength. YSA and DYA are in units of radians. The upper limit of grid 1 (XSA(2) = XSA(3)) and the upper limit of grid 2 (YSA(3)) may be changed and the densities of points may be changed. Boundaries that are zero should not be changed without modifying subroutine INTRP in NEC. The three grids must cover the region $0 \le R_1/\lambda \le 1$ and $0 \le \theta \le \pi/2$.

3. ARRAY DIMENSION LIMITATIONS

Number of Points in Interpolation Grids

Arrays:

COMMON/GGRID/AR1 (N_1, M_1, 4), AR2 (N_2, M_2, 4), AR3 (N_3, M_3, 4)

where $N_i \geq$ NXA(i) and $M_i \geq$ NYA(i)

The dimensions in common /GGRID/ in SOMNEC must be the same as the dimension of /GGRID/ in NEC.

Maximum Number of Iterations in GSHANK

Arrays:

Subroutine GSHANK: Q1 (6, MAXH), Q2 (6, MAXH)

where MAXH = maximum value of INT in GSHANK set at GS 13.

4. SOMNEC SUBROUTINE LINKAGE

Figure 17 shows the organization of subroutines in SOMNEC.

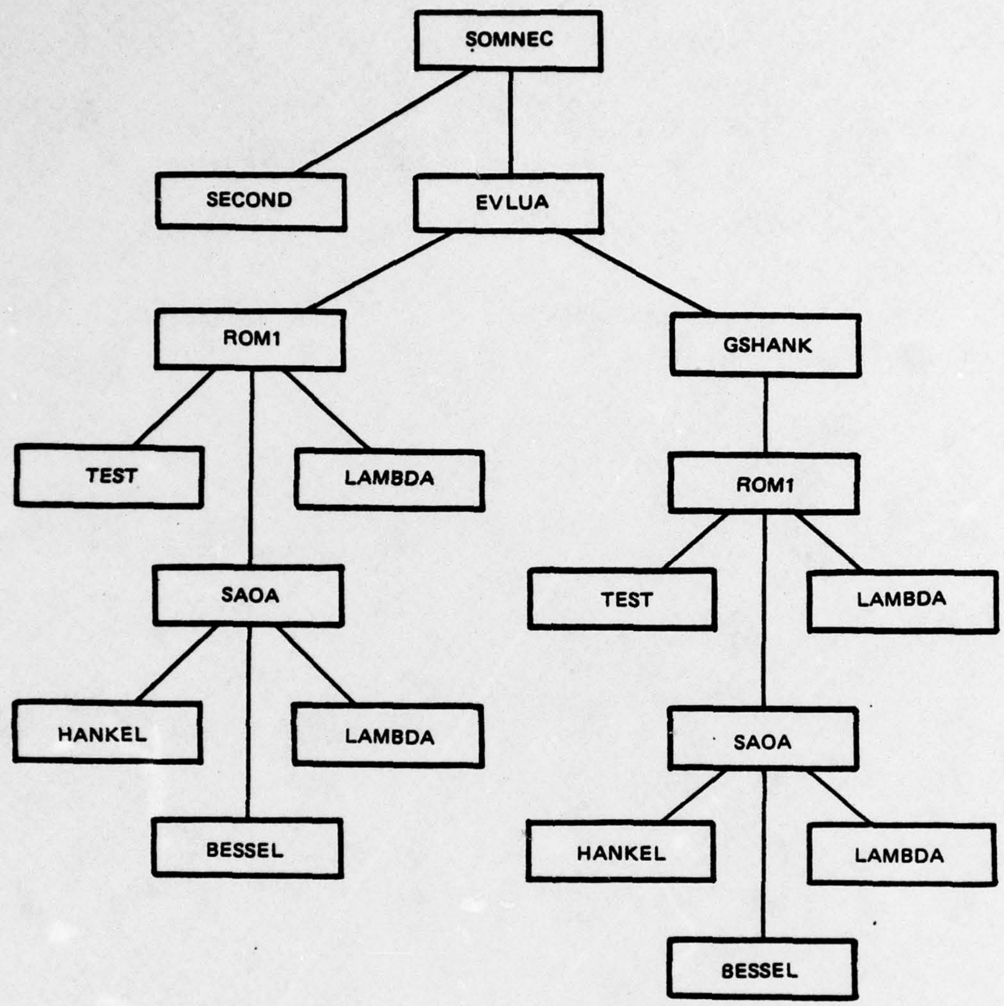


Figure 17. SOMNEC Subroutine Linkage Chart.

References

1. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill, New York, 1965.

2. Norton, K. A., "The Propagation of Radio Waves Over the Surface of the Earth and in the Upper Atmosphere," Proceedings of the Institute of Radio Engineers, Vol. 25, No. 9, Sept. 1937.

3. Miller, E. K., "A Variable Interval Width Quadrature Technique Based on Romberg's Method," Journal of Computational Physics, Vol. 5, No. 2, April 1970.

4. Miller, E. K., and G. J. Burke, "Numerical Integration Methods," IEEE Transactions, Vol. Ap-17, No. 5, Sept. 1969.

5. Handbook of Mathematical Functions, edited by M. Abramowitz, National Bureau of Standards (U.S.), Applied Mathematics Series 55, 1964.

6. Shanks, D., "Non-Linear Transformations of Divergent and Slowly Convergent Sequences," J. Math. Phys. 24, 1, 1955.

END
10-81
DTIC